الجمهورية العربية السورية

جامعة تشرين

كلية العلوم

قسم الرياضيات

تطبيقات برمجية على الشبكات

رسالة مقدمة لنيل درجة الماجستير في الرياضيات

إعداد

عبد القادر شيخ إبراهيم

بإشراف

أ.د. **أحمد الغصين**

١٤٣٠

٢٠٠٩

الأستاذ الدكتور عميد كلية العلوم

بعد اطلاع لجنة الحكم المشكلة لمناقشة رسالة الماجستير في الرياضيات التي قدمها الطالب عبد القادر شيخ إبراهيم بعنوان **تطبيقات برمجية على الشبكات** والمؤلفة من السادة:

الأستاذ الدكتور محمد خير أحمد

الأستاذ الدكتور أحمد الغصين

الدكتور محمد أسعد

على الرسالة المذكورة، فإننا نحيطكم علماً أن الطالب **عبد القادر شيخ إبراهيم** قد قام بإنجاز التوصيات العلمية والفنية التي أوصت بها لجنة الحكم.

نرجو الإطلاع واتخاذ ما ترونه مناسباً.

أ.د. محمد خير أحمد | أ.د. أحمد الغصين | د. محمد أسعد

الإهداء

إلى أمي الغالية و أبي الحبيب ...

إلى زوجتي الوفية و ابنتي الذكية ...

إلى أخوتي الأعزاء ...

أحبائي... إليكم جميعا أهدي عملي هذا.

شكر خاص

الشكر لله سبحانه وتعالى، ثم إلى المكان الذي احتضن انطلاقة مسيرتي في البحث العلمي وشهد ختامها، إلى جامعة تشرين متمثلة بالأستاذ الدكتور رئيس الجامعة صاحب الدور الكبير في دعم تقدم البحث العلمي أتوجه بخالص شكري و امتناني.

كما أشكر كلية العلوم ممثلة بالأستاذ الدكتور عميد الكلية، وإلى جميع أعضاء الهيئة التدريسية فيها على دعمهم لي و لزملائي الطلاب في مسيرتهم الدراسية، ولإسهامهم بدفع البحث العلمي في بلدنا إلى آفاق جديدة.

و إلى قسم الرياضيات خصوصاً، أتوجه بجزيل الشكر والامتنان والعرفان بالجميل، وبالأخص إلى أستاذي ومعلمي الأستاذ الدكتور أحمد الغصين، لك مني خالص الاعتراف بالفضل وأدعوه تعالى أن يوفقك إلى مزيد من التقدم في مسيرة البحث العلمي التي نذرت حياتك لأجلها، وأن ييسر لك السبل لكي تأخذ بأيدي المزيد من طلاب العلم ليسيروا على خطاك، إنه ولي ذلك والقادر عليه، وهو الموفّق.

تطبيقات برمجية على الشبكات

ملخص

تتعلق هذه الرسالة بدراسة شبكات القواسم و شبكات المجموعات الجزئية لإنشاء مخطط هاس لها و تحديد المثاليات و المرشحات فيها، و بدراسة المجموعات المرتبة جزئياً لمعرفة هل هي شبكة أم لا و لتحديد المثاليات و المرشحات فيها، وذلك عن طريق كتابة برامج حاسوبية مختصة بدراسة الشبكات.

وضعنا في هذا البحث ثلاثة تطبيقات برمجية متكاملة، التطبيق الأول مختص بإنشاء مخطط شبكة قواسم عدد ما، و يستطيع المستخدم تعديل شكل المخطط الناتج عن طريق سحب العناصر وإفلاتها، ويحدد المثاليات والمرشحات والشبكات الجزئية بلون مختلف ضمن المخطط.

التطبيق الثاني هو برنامج مشابه في المظهر والوظائف لبرنامج شبكات القواسم غير أنه مختص بشبكات المجموعات الجزئية في مجموعة منتهية. وفي هذا البرنامج حللنا مشكلة تمثيل المجموعات وعلاقة الاحتواء عن طريق استخدام العمليات المنطقية البسيطة.

التطبيق الثالث هو البرنامج الشامل الذي يقوم بدراسة أية مجموعة مرتبة لمعرفة فيما إذا شبكة أم لا. لإنجاز هذا البرنامج قدمنا مبرهنتين أساسيتين بوساطتهما يتم إيجاد الحدين الأعلى الأصغري والأدنى الأعظمي لأي عنصرين في المجموعة المرتبة جزئياً عن طريق حساب تقاطعات لمجموعات جزئية في المجموعة المرتبة المعطاة:

مبرهنة ١. لتكن المجموعة $S_{x,y} = U(x,y) \cap (\bigcap_{z \in U(x,y)} (z])$ عندئذ إما $S_{x,y} = \varnothing$ أو هي مجموعة تحوي عنصراً وحيداً هو $Sup(x,y)$.

مبرهنة ٢. لتكن المجموعة $N_{x,y} = L(x,y) \cap (\bigcap_{z \in L(x,y)} [z))$ عندئذ إما $N_{x,y} = \varnothing$ أو هي مجموعة تحوي عنصراً وحيداً هو $Inf(x,y)$.

حيث $U(x,y)$ ، $L(x,y)$ هما مجموعة الحدود العليا و مجموعة الحدود الدنيا على الترتيب للعنصرين x و y.

إن هذه التطبيقات البرمجية تيسر للباحث أو المدرس أو الطالب دراسة المفاهيم التي ذكرناها في نظرية الشبكات وتختصر الوقت اللازم لذلك.

فهرس المحتويات

٤§ الفصل الرابع

دراسة المجموعات المرتبة

مقدمة

تطبيقات برمجية على الشبكات

سميت البنى الجبرية باسمها لأنها تبنى بعضها فوق بعض، فلا حديث عن الحلقة مثلاً من دون الحديث عن الزمرة. وفي سعينا لإشراك الحاسوب في دراسة هذه البنى المجردة نحاول أن نجعل هذه الآلة العجماء تفهم على الأقل أبسط هذه البنى وهي المجموعة التي يمكن تعريفها بمعرفة العناصر التي تنتمي إليها. و لعل العلاقات بين العناصر في المجموعة أهم ما تجب دراسته عند الحديث عنها، لأنها حجر أساس في بناء العديد من البنى الجبرية.

ومن أهم أنواع العلاقات التي قد تتصف بها المجموعة، علاقة الترتيب الجزئي. فبالانطلاق منها يمكن تشكيل زمرة، وانطلاقاً من الزمرة يمكن تشكيل حلقة، وبالتالي إذا أدخلنا مفهوم المجموعات المرتبة إلى الحاسوب نكون قد خطونا الخطوة الأولى تجاه بناء تطبيقات برمجية قادرة على دراسة مختلف أنواع البنى الجبرية.

لا يخفى على أحد تقدم وسرعة إجراء الحسابات وتقييم الدوال باستخدام الحاسوب لذلك نجد الكثير من البرامج المختصة بالتحليل الرياضي وحساب كل شيء من أبسط الدوال حتى أعقد التكاملات بينما لا نجد اهتماماً مماثلاً بدراسة البنى الجبرية المجردة باستخدام الحاسوب.

اخترنا المجموعات المرتبة جزئياً والشبكات كنقطة بداية لدراسة البنى الجبرية باستخدام الحاسوب لأنها ذات صلة وثيقة بالبنى الجبرية الأخرى كالزمرة وجبر بول وبفروع الرياضيات الأخرى كالتبولوجيا. ولأنه يمكن تمثيلها وفق مخطط خاص يوضح العلاقة بين عناصرها.

إن الهدف من هذه الرسالة وضع تطبيقات برمجية متكاملة بعضها مع بعض، تمكن المستخدم من دراسة وتمثيل مخططات بعض الشبكات. ولقد استطعنا وضع ثلاثة تطبيقات تمكن الدارس من الاستغناء عن الرسم اليدوي لمخططات الشبكات وعن البرهان التقليدي لأن مجموعة مرتبة ما هي شبكة.

هذه التطبيقات مختصة بدراسة شبكتين شهيرتين هما شبكة القواسم وشبكة المجموعة الجزئية، ودراسة أية مجموعة مرتبة جزئياً لتمثيل مخططها ومعرفة هل هي شبكة أم لا، بالإضافة إلى استخراج المجموعات المرتبة الجزئية وتمثيل مخططها ضمن مخطط المجموعة الأساسية بلون مختلف. كما أن المخطط الناتج عن هذه التطبيقات هو مخطط تفاعلي، أي يتفاعل مع المستخدم عند نقر عناصره أو سحبها مما يحسن تجربة استخدام البرنامج.

ولإنشاء هذه التطبيقات البرمجية تم استخدام لغة Microsoft Visual Basic 6 لأنها تؤمن لنا ما نحتاجه من أدوات لإنشاء مخطط الشبكات، إلى جانب كونها واسعة الانتشار وعالية المستوى وسهلة الاستخدام تمكن المبرمج من إنشاء تطبيقات قوية من دون الخوض في تعقيدات أنظمة التشغيل.

تقسّم هذه الرسالة إلى أربعة فصول كالتالي: يوضح **الفصل الأول** المفاهيم الأساسية التي لا غنى عنها عند الحديث عن الشبكات، فأوردنا بعض التعاريف الأساسية و التمهيدات التي نحتاجها، بالإضافة إلى تعريف شبكتي القواسم و المجموعات المنتهية وطريقة تمثيلهما.

أما في **الفصل الثاني** فقد قدمنا برنامجاً أولياً يقوم بإنشاء مخطط شبكة قواسم عدد ما بعد إدخال العوامل الأولية لهذا العدد. ثم طورنا هذا التطبيق ليصبح بالإمكان إدخال العدد مباشرة بدلاً من عوامله الأولية، و ليستطيع المستخدم تعديل شكل المخطط الناتج عن طريق سحب العناصر وإفلاتها، وليكون قادراً على تحديد المثاليات والمرشحات والشبكات الجزئية بلون مختلف ضمن المخطط.
وتجدر الإشارة إلى أنه تم قبول بعض نتائج هذا الفصل للنشر في جامعة تشرين للدراسات والبحوث العلمية – سلسلة العلوم الأساسية
بعنوان /إنشاء مخطط هاس لشبكات القواسم برمجيا/

وتطويراً للبرنامج السابق قدمنا في **الفصل الثالث** برنامجاً مشابهاً في المظهر والوظائف لبرنامج شبكات القواسم غير أنه مختص بشبكات المجموعات المنتهية. وفي هذا البرنامج حللنا مشكلة تمثيل المجموعات وعلاقة الاحتواء عن طريق استخدام العمليات المنطقية البسيطة.

وقد خصصنا **الفصل الرابع** لكتابة البرنامج الشامل الذي يقوم بدراسة أية مجموعة مرتبة منتهية لمعرفة كونها شبكة أم لا. حيث بدأنا أولا بوضع أسس لعملنا عن طريق تحليل

متطلباته بشكل مفصل، ثم انتقلنا إلى كتابة صنفين برمجيين مختصين بدراسة المجموعات المنتهية والمجموعات المرتبة جزئياً والعمليات عليها.

ومن نتائج هذا الفصل أننا قدمنا مبرهنتين أساسيتين تفيدان في إيجاد الحدين الأعلى الأصغري والأدنى الأعظمي لأي عنصرين في المجموعة المرتبة جزئياً في حال وجودهما و ذلك عن طريق حساب تقاطعات لمجموعات جزئية في المجموعة المرتبة المعطاة، أي رددنا هذه المسألة إلى جبر المجموعات، حيث اعتمدنا بشكل كبير على الاجتماع والتقاطع والفرق في الخوارزميات التي قدمناها. ولذلك وضعنا في هذا البرنامج إجراءات خاصة بحساب نواتج هذه العمليات عن طريق عمليات منطقية بسيطة.

أضفنا إلى هذه البرامج إمكانية سحب العناصر وإفلاتها وتحديد مجموعات جزئية بلون مختلف، وإمكانية حفظ النتائج في ملف لمعاينتها و تعديلها لاحقا.

وفي نهاية الفصل عرضنا النتائج التي حصلنا عليها من البرنامج السابق، وتجدر الإشارة إلى أن القرص المدمج المرفق بهذه الرسالة يحوي العديد من الأمثلة المحفوظة ويمكن عرضها وتعديلها، بالإضافة إلى الشيفرة البرمجية الكاملة للتطبيقات الثلاثة التي قدمناها.

§١ الفصل الأول

تعريفات ومفاهيم أساسية
Basic Definitions and Concepts

١. المجموعات المرتبة جزئياً Partly Ordered Sets

نورد في هذا الفصل المفاهيم و التعريفات و المبرهنات الأساسية التي تُعدُّ ركيزة للفصول اللاحقة.

١-١ **تعريف.** لتكن A و B مجموعتين غير خاليتين. إن *العلاقة* $\mathfrak{R}:A \to B$ هي مجموعة جزئية من $A \times B$. وندعو العلاقات من A إلى A نفسها بعلاقات على A اختصاراً. إذا كان $\mathfrak{R} \ni (a, b)$ فإننا نكتب $a \,\mathfrak{R}\, b$ ونقول إن " a مرتبط بـ b وفق العلاقة $\mathfrak{R}$ ". أيضاً إذا كان a لا يرتبط بـ b وفق العلاقة $\mathfrak{R}$ فإننا نكتب $a \not\mathfrak{R}\, b$ [1].

إن أية علاقة $\mathfrak{R}$ على مجموعة غير خالية A قد تتصف بإحدى الخصائص التالية[1]:

$\mathfrak{R}$ *انعكاسية* (reflexive) إذا تحقق من أجل كل عنصر a في المجموعة A أن $a \,\mathfrak{R}\, a$.

$\mathfrak{R}$ *تناظرية* (symmetric) إذا تحقق من أجل كل عنصرين a و b في A أن $a \,\mathfrak{R}\, b \Leftarrow b \,\mathfrak{R}\, a$.

$\mathfrak{R}$ *تخالفية* (antisymmetric) إذا تحقق من أجل كل عنصرين a و b في A أن $a \,\mathfrak{R}\, b$ و $b \,\mathfrak{R}\, a \Leftarrow a = b$.

$\mathfrak{R}$ *متعدية* (transitive) إذا تحقق من أجل كل ثلاثة عناصر a و b و c في A أن $a \,\mathfrak{R}\, b$ و $b \,\mathfrak{R}\, c \Leftarrow a \,\mathfrak{R}\, c$.

١-٢ **تعريف.** تدعى العلاقة $\mathfrak{R}$ على المجموعة A بعلاقة ترتيب جزئي إذا كانت $\mathfrak{R}$ انعكاسية و تخالفية و متعدية[1].

وفي هذه الحالة ندعو $(A, \Re)$ بمجموعة مرتبة جزئياً (poset). نرمز لعلاقات الترتيب الجزئي عادةً بالرمز $\leq$ أو $\subseteq$ بدلا من $\Re$ و نقرأ أصغر أو يساوي أو محتواة في.

١-٣ أمثلة.

١) مجموعة الأعداد الطبيعية N هي مجموعة مرتبة بالنسبة لعلاقة الترتيب الطبيعية $x \leq y$.

٢) المجموعة $N^*=N\setminus\{0\}$ تشكل مجموعة مرتبة جزئياً بالنسبة لعلاقة يقسم $x \mid y$.

٣) المجموعة $P(E)$ هي مجموعة مرتبة جزئياً بالنسبة لعلاقة الاحتواء $X \subseteq Y$، حيث $P(E)$ هي أسرة كل المجموعات الجزئية في المجموعة E.

١-٤ تعريف. نقول عن العنصر a إنه أصغر تماماً من b (أو محتوى تماماً في b) ونكتب $a < b$ إذا كان $a \leq b$ و $a \neq b$ [2].

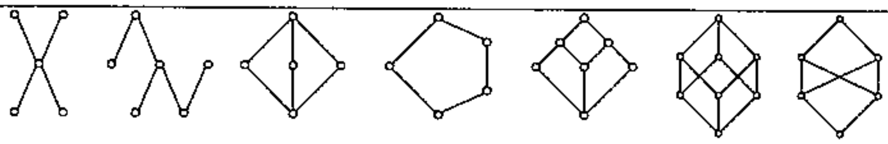
١-٥ تمهيدية. في أية مجموعة مرتبة، لا يوجد أي عنصر a يحقق $a < a$. بينما إذا كان $a < b$ و $b < c$ فإن ذلك يقتضي أن $a < c$. وبالعكس، إذا حققت أية علاقة $<$ الشرطين السابقين فإن تعريف العلاقة $a \leq b$ بالشكل: إما $a < b$ أو $a = b$، يعطينا علاقة ترتيب جزئي[3].

١-٥ تعريف. نقول إن " b يغطي a " في المجموعة المرتبة جزئياً A، إذا كان $a < b$ ولا يوجد أي عنصر $x \in A$ بحيث $a < x < b$ [3].

إذا كان عدد عناصر المجموعة المرتبة جزئياً A محدوداً، فإننا ندعو A مجموعة منتهية مرتبة جزئياً. و باستخدام علاقة يغطي يمكننا الحصول على تمثيل بياني (graphical representation) لأية مجموعة مرتبة منتهية A كما يلي[3]:

نرسم نقاطاً أو دوائرَ صغيرة لتمثيل كل عنصر من A، ونضع b فوق a إذا كان $a < b$. ثم نرسم خطاً مستقيماً من a إلى b إذا كان b يغطي a. إن المخطط الناتج يدعى بمخطط هاس (Hasse Diagram) للمجموعة المرتبة جزئياً A، و يظهر الشكل ١ بعض هذه المخططات.

إن $b > a$ إذا و فقط إذا أمكن التحرك نزولاً من b إلى a على طول خط في المخطط، ومن الواضح أنه يمكن أن يتم تعريف تقابل (isomorphism) بين أية مجموعة منتهية مرتبة جزئياً و مخطط هاس لهذه المجموعة.



الشكل ١: أمثلة على المخططات

٢. عناصر مميزة في المجموعات المرتبة جزئياً
Special Elements in Posets

٢-١ تعريف. إن *العنصر الأكبر* (greatest element) لأية مجموعة جزئية X من مجموعة مرتبة جزئياً A هو العنصر $g \in X$ الذي يحقق أن $g \geq x$ من أجل كل عنصر x من X. كما أن *العنصر الأصغر* (least element) للمجموعة X هو العنصر $l \in X$ الذي يحقق أن $l \leq x$ من أجل كل عنصر x من X [2].

٢-٢ تعريف. إن *الحد الأعلى* (upper bound) للمجموعة الجزئية X من مجموعة مرتبة جزئياً A هو العنصر $p \in A$ الذي يحقق أن $x \leq p$ من أجل كل عنصر x في X. كما أن *الحد الأدنى* (lower bound) للمجموعة X هو العنصر $q \in A$ الذي يحقق أن $q \leq x$ من أجل كل عنصر x في X [4].

٢-٣ مثال. لتكن المجموعة $A = \{2, 3, 4, 9\}$ المحتواة في المجموعة المرتبة جزئياً $(N^*, |)$ عندئذ يكون العدد 36 حداً أعلى للمجموعة A، و كذلك العدد 72، الخ... بينما يكون العدد 1 حداً أدنى للمجموعة A.

٢-٤ تعريف. *إن الحد الأعلى الأصغري* (least upper bound) للمجموعة X المحتواة في المجموعة المرتبة جزئيا A هو العنصر $s \in A$ الذي يحقق الشرطين:

- العنصر s هو حد أعلى للمجموعة X.
- إذا وجد حد أعلى آخر p فإن $s \leq p$. و نرمز $s = \sup(X)$.

كما *أن الحد الأدنى الأعظمي* (greatest lower bound) للمجموعة X هو العنصر $i \in A$ الذي يحقق الشرطين:

- العنصر i هو حد أدنى للمجموعة X.
- إذا وجد حد أدنى آخر q فإن $q \le i$. و نرمز $i = \inf(X)$ [4].

ينتج من التعريف أنه ليس بالضرورة وجود الحدين الأدنى الأعظمي و الأعلى الأصغري، ولكن إن وجدا فهما وحيدان.
فمثلا المجموعة A المعرفة في المثال (١§٢-٣) يكون العدد 36 هو الحد الأعلى الأصغري وأن العدد 1 هو الحد الأدنى الأعظمي.

٢-٥ تعريف[2]. بفرض $(A, \le)$ مجموعة مرتبة جزئياً و $a \in A$، عندئذ نعرف المجموعتين (a] و [a) على الشكل:

$$(a] = \{x \in A; x \le a\}$$
$$[a) = \{x \in A; a \le x\}$$

وسنستعمل هاتين المجموعتين بشكل كبير في دراستنا اللاحقة.

٣. الشبكات Lattices

تُعدُّ الشبكات من البنى الجبرية الهامة لارتباطها الوثيق بالزمر و التبولوجيا و المنطق الرياضي و جبر بول. ونورد فيما يلي بعض التعريفات والتمهيديات التي نحتاجها خلال هذا البحث.

٣-١ تعريف. إن *الشبكة* (lattice) هي مجموعة مرتبة جزئياً L بحيث يوجد لكل زوج من عناصرها x و y حد أدنى أعظمي أو "تقاطع"، و حد أعلى أصغري أو "اجتماع". ونرمز لهما $x \wedge y$ و $x \vee y$ على الترتيب [3].

٣-٢ تمهيدية. بفرض L شبكة و $x, y \in L$ عندئذ الشرط اللازم و الكافي ليكون $x \le y$ هو أن يكون $x \vee y = y$. أو أن يكون $x \wedge y = x$ [1].

٣-٣ نتيجة. بفرض L شبكة و $x, y \in L$ يكون $x \le x \vee y$ و $x \wedge y \le x$.

٣-٤ تعريف. بفرض L شبكة و $a, b \in L$، يمكن أن نعرف المجموعة $[a, b]$ على الشكل:

$$[a, b] = \{x \in L; a \wedge b \le x \le a \vee b\}$$

في هذه الحالة تكون المجموعة $[a, b]$ شبكة ندعوها الشبكة الجزئية المولدة بالعنصرين a و b.

٣-٥ تعريف[2]. نقول عن المجموعة الجزئية غير الخالية I من شبكة ما L أنها *مثالي* (ideal) في الشبكة L إذا كان:

- $x \le y, y \in I \Rightarrow x \in I$
- $x, y \in I \Rightarrow x \vee y \in I$

فمثلا تشكل المجموعة $(a]$ (انظر ١§٢-٥) مثالياً في الشبكة L من أجل كل عنصر a في L. و المثاليات من هذا الشكل تدعى بالمثاليات الرئيسة.

٣-٦ تعريف[2]. إن *المرشحة* (filter) هي مجموعة جزئية غير خالية F من شبكة ما L تحقق أن:

- $x \le y, x \in F \Rightarrow y \in F$
- $x, y \in F \Rightarrow x \wedge y \in F$

و تشكل المجموعة $[a)$ (انظر ١§٢-٥) مرشحة في الشبكة L من أجل كل عنصر a في L. وأيضاً تدعى المرشحات من هذا الشكل بالمرشحات الرئيسة.

٣-٧ تمهيدية. في الشبكات المنتهية يكون كل مثالي رئيساً، و كل مرشحة رئيسة.

و بما أن دراستنا تقتصر على الشبكات المنتهية فإن هذه التمهيدية تعطينا شكل المثاليات والمرشحات في هذه الشبكات.

٤. بعض أنواع الشبكات Some Lattices Types

نذكر في هذه الفقرة نوعين من الشبكات سيشكلان اللبنة الأساسية في دراستنا، وهما شبكة قواسم عدد صحيح و شبكة المجموعات الجزئية في مجموعة منتهية.

٤-١ تعريف. ليكن n عدداً طبيعياً و لتكن $D(n)=\{m\in N : m \mid n\}$ هي مجموعة القواسم الموجبة للعدد n، عندئذ تكون $(D(n), \mid)$ شبكة ندعوها *بشبكة قواسم* (Divisors Lattice) العدد n. وفي هذه الحالة يتحقق من أجل كل قاسمين n_1 و n_2 للعدد n ما يلي:

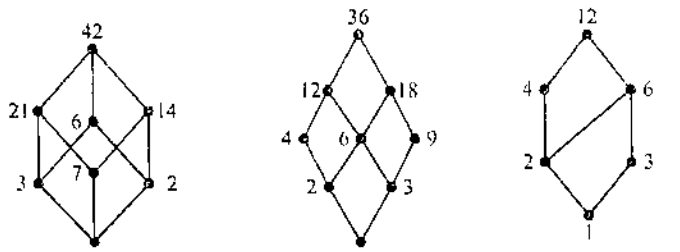
- $n_1 \vee n_2 = l.c.m(n_1, n_2)$
- $n_1 \wedge n_2 = g.c.d(n_1, n_2)$

حيث $l.c.m(n_1, n_2)$ و $g.c.d(n_1, n_2)$ هما على الترتيب المضاعف المشترك الأصغر و القاسم المشترك الأكبر للعددين n_1, n_2.

٤-٢ مثال. لنأخذ $D(12)=\{1, 2, 3, 4, 6, 12\}$ عندئذ $(D(12), \mid)$ شبكة فيها ست مثاليات وست مرشحات كل منها رئيسي، و يمكن رسم مخططها كما في الشكل ٢. كما يمكن تحديد المرشحة (2] كما في الشكل ٣.

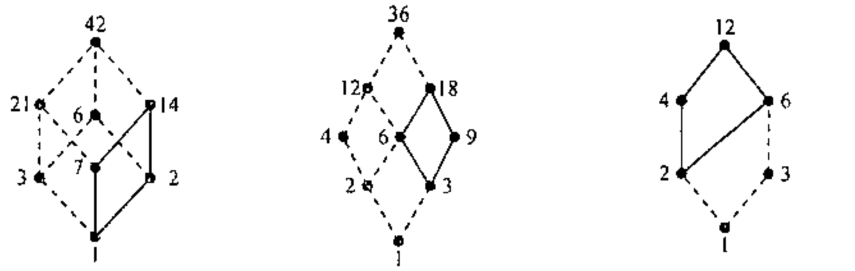
٤-٣ مثال. لنأخذ $D(36)=\{1, 2, 3, 4, 6, 9, 12, 18, 36\}$ عندئذ أيضاً $(D(36), \mid)$ شبكة فيها تسع مثاليات و تسع مرشحات يمكن رسم مخططها كما في الشكل ٢. ويمكن تحديد الشبكة الجزئية [6, 9] كما في الشكل ٣.

٤-٤ مثال. لنأخذ $D(42)=\{1, 2, 3, 6, 7, 14, 21, 42\}$ عندئذ $(D(42), \mid)$ شبكة يمكن رسم مخططها كما في الشكل ٢. و يمكن تحديد المثالي [14) كما في الشكل ٣.



$(D(42), \mid)$ $(D(36), \mid)$ $(D(12), \mid)$

الشكل ٢: أمثلة على شبكات القواسم



المثالي [14) في الشبكة (D(42), |) الشبكة الجزئية [6, 9] في الشبكة (D(36), |) المرشحة (2] في الشبكة (D(12), |)

الشكل ٣: أمثلة على مثالي و مرشحة و شبكة جزئية في شبكات القواسم

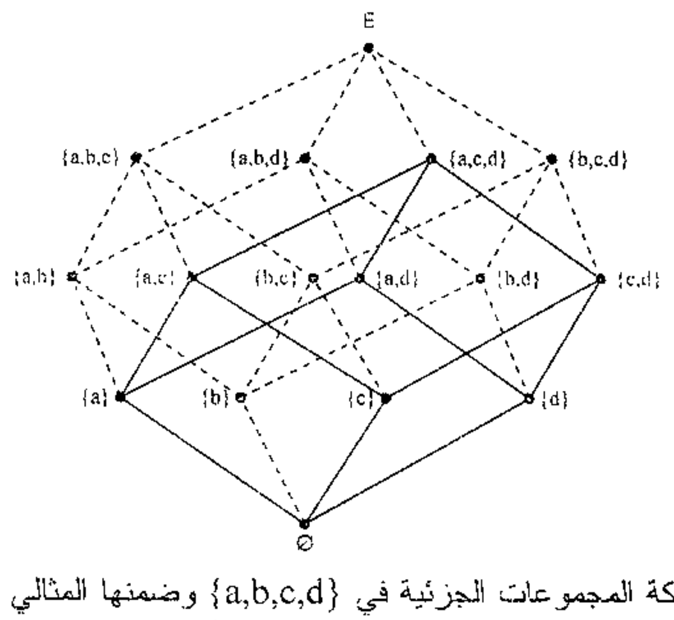
٤-٥ **تعريف.** لتكن E مجموعة منتهية، ولتكن $P(E)=\{D : D \subseteq E\}$ هي أسرة كل المجموعات الجزئية في E، عندئذ تكون $(P(E), \subseteq)$ شبكة ندعوها بـ*شبكة المجموعات الجزئية* في E. ويتحقق من أجل أي مجموعتين جزئيتين D_1 و D_2 في E ما يلي:

- $D_1 \vee D_2 = D_1 \cup D_2$
- $D_1 \wedge D_2 = D_1 \cap D_2$

٤-٦ **مثال.** إذا كانت $E=\{a, b, c, d\}$ عندئذ:

$$P(E)=\{\varnothing,\{a\},\{b\},\{c\},\{d\},\{a,b\},\{a,c\},\{b,c\},\{a,d\},\{b,d\},\{c,d\},\{a,b,c\},\{a,b,d\},\{a,c,d\},\{b,c,d\},E\}$$

و تكون $(P(E), \subseteq)$ شبكة يمكن رسم مخططها و تحديد المثالي [{a,c,d}) فيها كما في الشكل ٤.



الشكل ٤: شبكة المجموعات الجزئية في {a,b,c,d} وضمنها المثالي [{a,c,d})

٢§ الفصل الثاني

إنشاء مخطط هاس لشبكات القواسم
Creating Hasse Diagram of Divisors Lattices

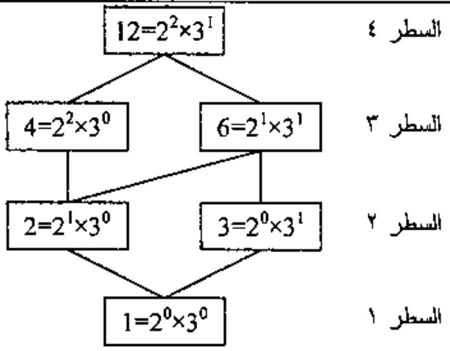
١. إنشاء مخطط الشبكة Creating the Diagram

نقدم أولا في هذا الفصل خوارزمية برمجية لإنشاء مخطط هاس لبعض شبكات القواسم، هذه الخوارزمية تولد تلقائيا مخطط شبكة قواسم العدد المطلوب، بعد إعطائها تحليل هذا العدد إلى عوامله الأولية. تُعدُّ هذه الخوارزمية خطوة هامة و بداية أساسية في كتابة تطبيقات برمجية قادرة على دراسة مختلف أنواع الشبكات و المجموعات المرتبة.

بدراستنا للعديد من شبكات القواسم و برسم مخطط هاس الذي يمثلها يدوياً وجدنا ثلاثة أمور تميز مخطط شبكة قواسم أي عدد صحيح:

- بما أن عناصر الشبكة هي قواسم للعدد المعطى فإن عواملها الأولية هي عوامل للعدد المعطى لكن بأسّات أصغر أو تساوي قوى العوامل الأولية للعدد و أكبر أو تساوي الصفر، أي يمكن الحصول على جميع قواسم العدد بأخذ الأعداد جميعها التي أسّات عواملها الأولية تحقق الشرط السابق.
- إن عناصر شبكة القواسم تتوضع على أسطر عددها يساوي مجموع قوى العوامل الأولية للعدد المعطى زائداً واحداً، بحيث إذا رقمنا الأسطر من الأسفل إلى الأعلى، فإن كل سطر يحوي القواسم التي مجموع قوى عواملها الأولية يساوي رقم السطر ناقصاً واحداً.
- إن عدد الخطوط التي تصل بين عنصر من سطر ما و بين عناصر السطر الذي تحته يساوي عدد عوامله الأولية التي قواها أكبر من الصفر.

كمثال بسيط على ذلك لنأخذ الشبكة $(D(12),|)$ كما في (الشكل ٥).



الشكل ٥: المخطط التفصيلي لشبكة قواسم العدد ١٢

نلاحظ أن السطر 1 يحوي الأعداد التي مجموع قوى عواملها الأولية صفر، أي يحوي العدد 1 فقط، والسطر 2 الذي يعلوه يحوي الأعداد التي مجموع قوى عواملها الأولية يساوي 1، أي يحوي العددين 2 ، 3. أما السطر 3 فيحوي الأعداد التي مجموع قوى عواملها الأولية يساوي 2 أي يحوي العددين 4 ، 6. وهكذا حتى الوصول إلى السطر الرابع (السطر الأعلى) حيث فيه جميع العناصر التي مجموع قوى عواملها الأولية يساوي مجموع قوى العوامل الأولية للعدد الذي نرسم شبكة قواسمه، أي يحوي العدد 12 فقط.

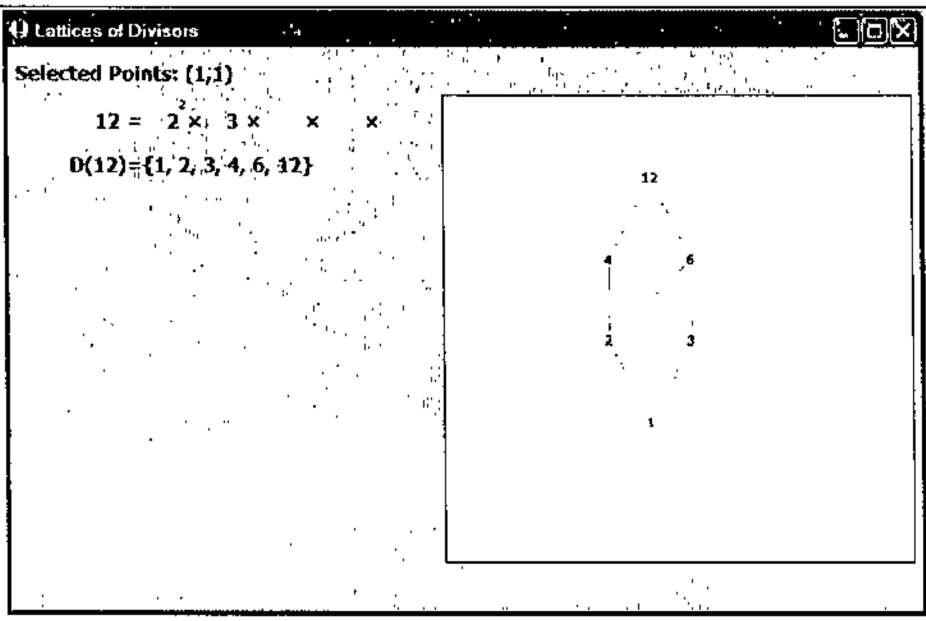
ومن جهة أخرى، إن عدد العوامل الأولية للعددين 6 ، 12 التي قواها أكبر من الصفر هو 2 لهذا يرتبط كل منهما بالسطر الذي تحته بخطين، بينما يرتبط العدد 4 بالسطر الذي تحته بخط واحد لأن له عاملاً أولياً واحداً قوته أكبر من الصفر، و الأمر ذاته ينطبق على العددين 2 و 3.

٢. تصميم واجهة البرنامج Designing Interface

في أول مرحلة من مراحل تصميم البرنامج، وضعنا على النافذة الرئيسة عشر مربعات نص (TextBox)[5] خمسة منها لإدخال عوامل العدد الأولية، و وضعنا فوق كل مربع نص منها مربعاً آخر لإدخال قوة كل عامل، صغرنا حجم الخط فيه، وغيرنا لون خلفية (BackColor) كل الصناديق ليماثل لون النافذة بحيث تبدو هذه الصناديق مدموجة بالنافذة. ثم فصلنا بين المربعات الخمسة الأولى بأربع لافتات (Label) ضبطنا خاصية العنوان لها لتظهر الرمز ×، و وضعنا قبل مربع النص الأول أداة لافتة لنضع فيها العدد الناتج عند تغيير نص أحد مربعات النصوص العشرة السابقة، و وضعنا مربع لوحة (PictureBox) يعطي مساحة بيضاء تشغل معظم الشاشة لإنشاء المخطط فيها.

بما أن المخطط مؤلف من مجموعة من الأعداد تصل بينها مجموعة من الخطوط، استخدمنا لإنشاء مخطط الشبكة أداة لافتة (Label) واحدة وضعناها على مربع اللوحة السابق، وضبطنا خاصية الفهرس (Index) بالعدد 0، ووضعنا مع أداة اللافتة أداة خط (Line) واحدة ضبطنا أيضاً خاصية الفهرس لها بالعدد 0 لكي نشكل مصفوفتي (Array) أدوات. سنستعمل اللافتة لتمثيل عناصر الشبكة و الخط لتمثيل العلاقة بين العناصر.

لا بد من الإشارة إلى أنه يمكن استخدام إجراءات الرسم الموجودة ضمن فيجوال بيسيك، لكن هدفنا إنشاء مخطط تفاعلي يُمكّن المستخدم من تحريك العناصر عن طريق سحبها و إفلاتها، و تحديد شبكة جزئية أو مثالي أو مرشحة بتغيير لون الخطوط الواصلة بين العناصر، وبما أن اللافتة تستجيب لأحداث الفأرة من نقر وسحب وإفلات، و بما أن مصفوفات الأدوات تساعد على كتابة إجراء يتعامل مع جميع عناصرها، و يمكن زيادة عدد هذه العناصر أو إنقاصه بحسب الحاجة، استعملنا الأدوات بدلاً من إجراءات الرسم. كما لابد أيضا من الإشارة إلى أن مربعات النصوص السابقة تسمح بإدخال خمسة عوامل أولية فقط للعدد المعطى، ويمكن زيادة عدد هذه المربعات عند الحاجة. هذه أول مرحلة من مراحل تصميم البرنامج، ورتبنا الواجهة بحيث تؤمن خمسة حقول لإدخال عوامل العدد و مساحة بيضاء تشغل معظم الشاشة لإنشاء المخطط فيها.



الشكل ٦: واجهة البرنامج بعد تشغيله و إعطائه الأمر برسم شبكة قواسم العدد ١٢.

٣. كتابة الشيفرة البرمجية Typing the Code

عند تغيير النص الموجود في أحد مربعات النصوص يقع الحدث Change [3]، ضمن إجراء هذا الحدث نضع عبارة لحساب العدد الناتج عن ضرب العوامل الأولية المدخلة في صناديق النصوص الخمسة السفلية بعد رفعها إلى القوى المدخلة في الصناديق العلوية، ولتخزين العدد مع عوامله الأولية نعرف نمطاً (Type)[3] جديداً بالشكل التالي:

```
Private Type Primes
   Prime(4) As Integer
   Expo(4) As Integer
   Number As Long
   Es As Integer
End Type
```

نخزن في المصفوفة Prime(4) العوامل الأولية الخمسة (ترقيم عناصر المصفوفة يبدأ من 0 لذلك وضعنا 4)، وفي المصفوفة Expo(4) قوى العوامل الأولية بالترتيب نفسه، وفي Number العدد نفسه، أما في Es فنخزن مجموع القوى السابقة لحاجتنا إليه في الخوارزمية وللاستغناء عن حسابه كلما احتجنا إليه لاحقاً.

و بعد ذلك نعلن عن المتغيرات العامة المطلوبة كما يلي:

```
Dim Num As Primes
Dim Divisors() As Primes
Dim Es() As Integer
Dim Lines As Integer
```

المتغير Num يمثل العدد وعوامله الأولية و يتم حسابه ضمن الحدث Change لمربعات النصوص العشرة، المصفوفة Divisors() هي مصفوفة ديناميكية (يمكن تحديد عدد عناصرها بحسب الحاجة) يمثل كل عنصر فيها أحد قواسم العدد المعطى مع عوامله الأولية، أما المصفوفة Es() فهي مصفوفة ديناميكية نحدد عدد عناصرها ليساوي عدد أسطر المخطط المطلوب، و يمثل العنصر ذو الترتيب i فيها عدد القواسم التي مجموع قوى عواملها الأولية هو العدد i، وبالنسبة للمتغير Lines فهو يمثل عدد العلاقات (الخطوط) اللازم تمثيلها.

و على ضوء ما أوردناه سابقاً في ٢§١، فإننا نكتب الإجراء الأساسي الذي يحسب قواسم العدد والمعلومات اللازم تخزينها في المتغيرات السابقة الذكر:

الشيفرة البرمجية الكاملة للإجراء GenerateDivisors

```
Private Sub GenerateDivisors()
Dim h%, i%, j%, k%, m%, l%

ReDim Divisors(0) 'نضبط حجم المصفوفة لتحوي عنصرا واحدا فقط
                  '(ترقيم المصفوفة يبدأ من ٠)
```

الشيفرة البرمجية الكاملة للإجراء GenerateDivisors

```
ReDim Es(Num.Es)       ' نضبط حجم المصفوفة ليساوي مجموع قواسم العدد
                       ' المعطى زائد واحد أي يساوي عدد أسطر المخطط
Lines = 0              ' نهيئ المتغير بالعدد ٠

For l = 0 To Num.Es          ' نكرر من أول سطر في المخطط إلى آخر سطر
                             ' (نبدأ من ٠ وحتى عدد الأسطر ناقص ١)
For m = 0 To Num.Expo(4) ' نولد القواسم عن طريق أخذ
For k = 0 To Num.Expo(3) ' العوامل الأولية للعدد
For j = 0 To Num.Expo(2) ' المعطى وإعطائها قوى تبدأ من ٠ و تنتهي
For i = 0 To Num.Expo(1) ' بقوة العامل في العدد المعطى ولأنه لدينا
For h = 0 To Num.Expo(0) ' خمسة عوامل أولية نكتب خمسة تكرارات متداخلة
  ' إذا كان مجموع القوى يساوي رقم السطر ناقص ١
  If h + i + j + k + m = l Then
    ' عوامل القاسم هي عوامل العدد المعطى نفسها
    Divisors(UBound(Divisors)).Prime(0) = Num.Prime(0)
    Divisors(UBound(Divisors)).Prime(1) = Num.Prime(1)
    Divisors(UBound(Divisors)).Prime(2) = Num.Prime(2)
    Divisors(UBound(Divisors)).Prime(3) = Num.Prime(3)
    Divisors(UBound(Divisors)).Prime(4) = Num.Prime(4)
    ' قوى العوامل الأولية نأخذها من التكرار
    Divisors(UBound(Divisors)).Expo(0) = h
    Divisors(UBound(Divisors)).Expo(1) = i
    Divisors(UBound(Divisors)).Expo(2) = j
    Divisors(UBound(Divisors)).Expo(3) = k
    Divisors(UBound(Divisors)).Expo(4) = m
    ' نحسب القاسم بأخذ جداء العوامل الأولية بعد رفعها إلى قواها
    Divisors(UBound(Divisors)).Number = _
      (Num.Prime(0) ^ h) * (Num.Prime(1) ^ i) * _
      (Num.Prime(2) ^ j) * (Num.Prime(3) ^ k) * _
      (Num.Prime(4) ^ m)
    ' نحسب مجموع القوى
    Divisors(UBound(Divisors)).Es = h + i + j + k + m
   ' نضيف ١ إلى عدد العوامل الأولية التي مجموع قواها يساوي رقم السطر ناقص ١
    Es(l) = Es(l) + 1
    ' نضيف إلى عدد الخطوط الكلي عدد الخطوط التي يرتبط بها
    ' هذا العنصر مع عناصر السطر الذي تحته
    Lines = Lines + Sgn(h) + Sgn(i) + Sgn(j) + _
                    Sgn(k) + Sgn(m)
    ' نضيف عنصر جديد إلى المصفوفة
    ReDim Preserve Divisors(UBound(Divisors) + 1)
  End If
Next
Next
Next
Next
Next
Next
' بعد الانتهاء يكون في نهاية المصفوفة عنصر فارغ زائد نحذفه
ReDim Preserve Divisors(UBound(Divisors) - 1)
End Sub
```

إن الإجراء السابق يولد قواسم العدد عن طريق أخذ العوامل الأولية للعدد و إعطائها قوى متزايدة من الصفر و حتى أكبر قوة للعامل الأولي التي تقسم العدد. أما لحساب المتغير

Lines ذكرنا أن كل قاسم في الشبكة يكون عدد العناصر التي ترتبط به من الأسفل يساوي عدد العوامل الأولية لهذا العنصر التي قواها تختلف عن الصفر. فإذا نظرنا إلى السطر

Lines = Lines + Sgn(h) + Sgn(i) + Sgn(j) + Sgn(k) + Sgn(m)

لوجدنا أنه يستخدم التابع Sgn [5] لتحديد إذا كانت القوة تساوي الصفر أو لا، معلوم أن التابع السابق يعيد 1- إذا كان العدد سالبا و 1 إذا كان العدد موجبا و 0 إذا كان العدد صفرا، فبدلا من استخدام خمس عبارات شرطية قمنا باستخدام هذا التابع.

الآن نكتب الإجراء الذي يقوم بإنشاء المخطط، و مهمته تتلخص بضبط عدد مصفوفة اللافتات و مصفوفة الخطوط ثم تحريك كل عنصر من عناصر المصفوفتين إلى موقعه المناسب، و من أجل ضبط المسافة بين عناصر الشبكة صرحنا عن الثابتين التاليين:

```
Const SPA = 1000        'المسافة بين كل عنصرين طوليا أو عرضياً
Const SPR = 2           'عامل تقسيم يحدد مسافة القفزة
```

فبتعديل هذين الثابتين يمكن تعديل المسافة بين عناصر المخطط من دون تعديل الشيفرة البرمجية للإجراء الخاص بتوليد مخطط هاس لشبكة القواسم المطلوبة و الذي نكتبه على النحو التالي:

الشيفرة البرمجية الكاملة للإجراء GenerateDivisors

```
Private Sub GenerateLattice()
Dim i%, j%, l%, Esp() As Integer
Dim LastS As Integer
'نضيف اللافتات اللازمة و نزيل الزائدة عن حاجتنا
For i = lblP.Count To UBound(Divisors)
    Load lblP(i)
Next
For i = UBound(Divisors) + 1 To lblP.Count - 1
    Unload lblP(i)
Next
'نضيف الخطوط اللازمة ونزيل الزائدة عن حاجتنا
For i = linL.Count To Lines - 1
    Load linL(i)
Next
For i = Lines To linL.Count - 1
    Unload linL(i)
Next
'عدد عناصر المصفوفة يساوي عدد الأسطر، كل عنصر من هذه المصفوفة يمثل عدد
'عناصر السطر التي تم ضبط موضعها
ReDim Esp(Num.Es)
'تهيئة المتغيرات بالعدد ٠
LastS = 0  'هذا المتغير يمثل آخر عنصر من عناصر السطر الذي تحت العنصر الحالي
l = 0
For i = 0 To lblP.Count - 1  'نكرر من أول لافتة إلى آخر لافتة
    Dim nX%, nY%
    'نحسب عدد القفزات العرضية ابتداءً من حافة مربع اللوحة اليسرى
    'و حتى العنصر المطلوب اعتماداً على عدد عناصر السطر الكلية
    'و عدد العناصر التي تم ضبط موقعها
    nX = Int((picRight.ScaleWidth / 2 + SPA / 2 -
```

الشيفرة البرمجية الكاملة للإجراء GenerateDivisors

```
        Es(Divisors(i).Es) / 2 * SPA + _
        Esp(Divisors(i).Es) * SPA) / (SPA / SPR))
    'نحسب عدد القفزات الطولية ابتداءً من الحافة العليا لمربع اللوحة
    ' و حتى السطر الذي ينتمي إليه العنصر اعتماداً على عدد الأسطر
    ' الكلية و على رقم السطر الذي ينتمي إليه العنصر
    nY = Int((picRight.ScaleHeight / 2 + _
        Num.Es / 2 * SPA - _
        Divisors(i).Es * SPA) / (SPA / SPR))
    'نضبط عنوان اللافتة لتحوي القاسم
    lblP(i).Caption = Trim(Divisors(i).Number)
    lblP(i).Tag = Trim(Divisors(i).Number)
    'نحرك اللافتة إلى موقعها المناسب اعتمادا على عدد القفزات
    ' الطولية و العرضية
    lblP(i).Move nX * (SPA / SPR) - lblP(i).Width / 2, _
                nY * (SPA / SPR) - lblP(i).Height / 2
    'نضيف ١ إلى عدد عناصر السطر التي تم ضبط موقعها ثم نظهر اللافتة
    Esp(Divisors(i).Es) = Esp(Divisors(i).Es) + 1
    lblP(i).Visible = True
    'إذا كان مجموع قوى العنصر مختلفاً عن الصفر فالعنصر مرتبط
    ' مع عناصر السطر الذي تحته
    If Divisors(i).Es Then
      'نكرر من أول عنصر من عناصر السطر الذي يقع تحت العنصر
      ' الحالي إلى آخر عنصر
      For j = LastS - Es(Divisors(i).Es - 1) + 1 To LastS
        'نختبر علاقة يقسم
        If Divisors(i).Number Mod Divisors(j).Number = 0 Then
          'العلاقة محققة ، نحرك أحد الخطوط ليصل بين العنصرين
          linL(l).X1 = lblP(j).Left + lblP(j).Width / 2
          linL(l).Y1 = lblP(j).Top + lblP(j).Height / 2
          linL(l).X2 = lblP(i).Left + lblP(i).Width / 2
          linL(l).Y2 = lblP(i).Top + lblP(i).Height / 2
          linL(l).Visible = True
          l = l + 1
        End If
      Next
      'إذا كان هذا العنصر آخر عنصر في السطر فاحتفظ برقمه لحاجتنا إليه
      If LastS + Es(Divisors(i).Es) = i Then LastS = i
    End If
    lblP(i).ZOrder  'اجعل ترتيب اللافتة فوق ترتيب باقي الأدوات
Next  'انتقل إلى العنصر التالي
End Sub
```

إن ()lblP هي مصفوفة اللافتات التي نستخدمها لتمثيل عناصر الشبكة، و ()linL هي مصفوفة الخطوط التي نستخدمها لتمثيل العلاقات بين العناصر، و picRight هو مربع اللوحة الذي يحوي مصفوفتي اللافتات و الخطوط، كما أسلفنا في الفقرة ٢.

في بداية الإجراء نقوم بالتخلص من اللافتات و الخطوط الزائدة عن طلبنا إذا كان هنالك زيادة، أما إذا لم توجد زيادة نقوم بإنشاء ما يلزمنا من لافتات و خطوط، وذلك من خلال حلقات التكرار الأربع الأولى. تجدر الإشارة إلى أنه يلزمنا عدد من اللافتات يساوي عدد العناصر في المصفوفة ()Divisors، وعدد من الخطوط يساوي العدد الموجود في المتغير

Lines، وتم حساب هذه المصفوفة و هذا المتغير في الإجراء GenerateDivisors الـذي أسلفنا ذكره.

لترتيب العناصر بالشكل الصحيح يبرز استخدام المصفوفة ()Es التي سـبق ذكرهـا، فمـن خلالها نحدد عدد العناصر في أي سطر. الآن ندخل في تكرار لضبط موقع و نص كل مـن اللافتات، فنضع في المتغيرين nX و nY عدد القفزات الطولية و العرضـية التـي تحـدد إحداثيات العنصر الحالي معتمدين على عدد العناصر في السطر الذي ينتمي إليه (موجود في المصفوفة Es) وعلى عدد العناصر التي تم ضبط موقعها قبله في السطر نفسه (موجود فـي المصفوفة Esp)، ثم نحرك اللافتة إلى المكان المطلوب عن طريق ضرب عدد القفزات بطول القفزة لنحصل على الإحداثيات، ونضبط عنوانها ليحوي العنصر الحالي.

قلنا سابقاً إن كل عنصر في الشبكة يرتبط من الأسفل بعدد من العناصر يساوي عدد عواملـه الأولية التي قواها مختلفة عن الصفر، لذلك نختبر القيمة Divisors(i).Es فإذا حوت قيمـة مختلفة عن الصفر فمجموع العوامل مختلف عن الصفر و بالتالي أحد العوامل قوته مختلفـة عن الصفر، فندخل في تكرار و نختبر جميع عناصر السطر الذي يقع تحت سطر العنـصر الحالي لنرى أيٌّ منها يقسم العنصر الحالي؟، ثم نقوم بتحريك أحد عناصر المصفوفة ()linL ليصل بين العنصر الحالي و العنصر الذي يقسمه. نقوم أخيرا بنقل اللافتة إلـى أمـام بـاقي الأدوات عن طريق الإجراء ZOrder [5] لنضمن أن لا يغطيهـا أيٌّ خـط مـن خطـوط المصفوفة ()linL.

٤. تحسين برنامج إنشاء مخطط هاس لشبكات القواسم
Enhancing the Program

عند اختبار البرنامج السابق حصلنا على نتائج مقبولة عند رسم مخططات لبعض الـشبكات، لكن لا بد من تطوير البرنامج ليتيح إمكانية تحريك عناصر المخطط مما يحسن شكله لإبراز جانب معين منه، مما يزيد من الفائدة العلمية و التطبيقية للبرنامج و لابد من أن يتم توسـيع نطاق الأعداد التي يمكن إنشاء شبكة قواسمها عن طريق إدخال العدد بدلا من إدخال العوامل الأولية.

٤-١ مبرهنـة. كل عدد صحيح $n > 1$ إما أن يكون أولياً و إما أن يقبل عاملاً أولياً p يحقق $p \leq \sqrt{n}$ [6].

المسألة الأولى في الأهمية هي تزويد البرنامج بخوارزمية خاصة بتحليل الأعداد إلى عواملها الأولية و بالاعتماد على المبرهنة السابقة نورد الخوارزمية التالية:

١. ليكن r العدد المطلوب تحليله

٢. $n \leftarrow 0$

٣. كرر مادام $2 \mid r$

أ. $n \leftarrow 1$

ب. $prim_0 \leftarrow 2$

ج. $exponent_0 \leftarrow exponent_0 + 1$

د. $r \leftarrow r / 2$

٤. $d \leftarrow 3$

٥. كرر مادام $d \leq \sqrt{r}$

أ. كرر مادام $d \mid r$

١. إذا $prim_n = d$ نفذ

أ. $exponent_n \leftarrow exponent_n + 1$

٢. و إلا

أ. $n \leftarrow n + 1$

ب. $prim_n \leftarrow d$

ج. $exponent_n \leftarrow 1$

٣. $r \leftarrow r / d$

ب. $d \leftarrow d + 2$

بعد تنفيذ هذه الخوارزمية سيحوي n عدد العوامل الأولية للعدد r و prim العوامل الأولية و exponent قوى هذه العوامل.

٤-٢ مبرهنة. بفرض n عدداً صحيحاً يكتب كجداء لقوى عوامل أولية بالشكل $n = p_1^{\varepsilon_1} . p_2^{\varepsilon_2} \ldots p_m^{\varepsilon_m}$ فإن دالة عدد قواسم العدد n و نرمز لها $\tau(n)$ تعطى بالشكل $\tau(n) = (\varepsilon_1 + 1)(\varepsilon_2 + 1)\ldots(\varepsilon_m + 1)$ [6].

و فيما يلي نورد الشيفرة البرمجية الكاملة للنموذج frmMain الذي يمثل البرنامج المحسن، حيث قمنا بتعديل الشيفرة لتسمح بتحديد شبكات جزئية، و تحريك عناصر المخطط بالإضافة إلى تحليل العدد إلى عوامله الأولية.

الشيفرة البرمجية الكاملة للنموذج frmMain

```
VERSION 5.00
Begin VB.Form frmMain
   Caption         =   "Lattices of Divisors"
   ClientHeight    =   7560
```

```
ClientLeft      =   60
ClientTop       =   555
ClientWidth     =   8850
ScaleHeight     =   7560
ScaleWidth      =   8850
StartUpPosition =   2  'CenterScreen
Begin VB.Frame fraTop
   BorderStyle     =   0  'None
   ForeColor       =   &H00000000&
   Height          =   1140
   Left            =   0
   TabIndex        =   2
   Top             =   0
   Width           =   8115
   Begin VB.CommandButton cmdDraw
      Caption         =   "Draw"
      Height          =   420
      Left            =   45
      TabIndex        =   15
      Top             =   675
      Width           =   1320
   End
   Begin VB.TextBox txtNumber
      Alignment       =   1  'Right Justify
      BorderStyle     =   0  'None
      Height          =   285
      Left            =   90
      TabIndex        =   12
      Text            =   "12"
      Top             =   270
      Width           =   1545
   End
   Begin VB.CommandButton cmdScrollDown
      Caption         =   "<"
      Height          =   240
      Left            =   4770
      TabIndex        =   11
      Top             =   360
      Width           =   240
   End
   Begin VB.TextBox txtDSet
      BackColor       =   &H8000000F&
      Height          =   510
      Left            =   5130
      Locked          =   -1  'True
      MultiLine       =   -1  'True
      ScrollBars      =   2  'Vertical
      TabIndex        =   10
      Top             =   90
      Width           =   2625
   End
   Begin VB.CommandButton cmdScrollUP
      Caption         =   ">"
      Height          =   240
      Left            =   4770
      TabIndex        =   9
      Top             =   90
      Width           =   240
   End
```

```
Begin VB.Frame fraFactor
   BackColor       =   &H80000005&
   BorderStyle     =   0  'None
   Height          =   510
   Left            =   1935
   TabIndex        =   4
   Top             =   90
   Width           =   2805
   Begin VB.Label lblPrime
      AutoSize        =   -1  'True
      BackStyle       =   0  'Transparent
      Caption         =   "2"
      Height          =   285
      Index           =   0
      Left            =   180
      TabIndex        =   8
      Top             =   180
      Width           =   150
   End
   Begin VB.Label lblExponent
      AutoSize        =   -1  'True
      BackStyle       =   0  'Transparent
      Caption         =   "2"
      Height          =   165
      Index           =   0
      Left            =   360
      TabIndex        =   7
      Top             =   45
      Width           =   75
   End
   Begin VB.Label lblX
      AutoSize        =   -1  'True
      BackStyle       =   0  'Transparent
      Caption         =   "x"
      Height          =   285
      Index           =   0
      Left            =   540
      TabIndex        =   6
      Top             =   180
      Width           =   180
   End
   Begin VB.Label lblFactor
      BackStyle       =   0  'Transparent
      BorderStyle     =   1  'Fixed Single
      Height          =   510
      Left            =   0
      TabIndex        =   5
      Top             =   0
      Width           =   2805
   End
End
Begin VB.Label Label2
   AutoSize        =   -1  'True
   BackStyle       =   0  'Transparent
   Caption         =   "="
   ForeColor       =   &H00000000&
   Height          =   285
   Left            =   1710
   TabIndex        =   14
```

```
         Top             =   270
         Width           =   195
      End
      Begin VB.Label lblNumber
         BackColor       =   &H80000005&
         BorderStyle     =   1  'Fixed Single
         Height          =   510
         Left            =   45
         TabIndex        =   13
         Top             =   90
         Width           =   1635
      End
      Begin VB.Label lblS
         AutoSize        =   -1  'True
         Caption         =   "Selected Points: (1,1)"
         Height          =   285
         Left            =   2835
         TabIndex        =   3
         Top             =   720
         Width           =   2640
      End
   End
   Begin VB.PictureBox picCon
      BackColor       =   &H00FFFFFF&
      Height          =   4470
      Left            =   45
      ScaleHeight     =   294
      ScaleMode       =   3  'Pixel
      ScaleWidth      =   429
      TabIndex        =   0
      Top             =   1170
      Width           =   6495
      Begin VB.Label lbl
         Alignment       =   2  'Center
         AutoSize        =   -1  'True
         BackColor       =   &H00FFFFFF&
         Caption         =   "#"
         ForeColor       =   &H00000000&
         Height          =   195
         Index           =   0
         Left            =   2295
         TabIndex        =   1
         Top             =   3105
         Visible         =   0   'False
         Width           =   150
      End
      Begin VB.Line lin
         BorderColor     =   &H00C0C0C0&
         Index           =   0
         Visible         =   0   'False
         X1              =   117
         X2              =   120
         Y1              =   216
         Y2              =   171
      End
   End
End
Attribute VB_Name = "frmMain"
Attribute VB_GlobalNameSpace = False
```

الشيفرة البرمجية الكاملة للنموذج frmMain

```
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit

' هذه المتغيرات مطلوبة لتحريك عناصر المخطط
Dim mF As Boolean
Dim mX As Single
Dim mY As Single

' النقطتان اللتان تم تحديدهما لتلوين الشبكة الجزئية المولدة بهما
Dim Point1 As Integer
Dim Point2 As Integer

' تم شرح الباقي مسبقاً
Const SPA = 100
Const SPP = 2

Private Type Primes
    Prime() As Long
    Exponent() As Integer
    Number As Long
    ES As Integer
End Type

Dim Num As Primes
Dim Divisors() As Primes
Dim ES() As Integer
Dim Lines As Integer

' عند النقر على زر الرسم
Private Sub cmdDraw_Click()
GenerateLattice ' استدعاء إجراء توليد الشبكة
picCon.SetFocus
End Sub

' عند النقر على زر تحريك العوامل الأولية نحو اليمين
Private Sub cmdScrollDown_Click()
' إذا كانت أول لافتة من جهة اليسار ظاهرة تماماً فاخرج
If lblPrime(0).Left > 0 Then Exit Sub
' حرك كل اللافتات نحو اليمين بمقدار ٢٠٠
Dim i As Integer
For i = 0 To lblX.Count - 1
    lblPrime(i).Left = lblPrime(i).Left + 200
    lblExponent(i).Left = iblExponent(i).Left + 200
    lblX(i).Left = lblX(i).Left + 200
Next
lblPrime(i).Left = lblPrime(i).Left + 200
lblExponent(i).Left = lblExponent(i).Left + 200
End Sub

' عند النقر على زر تحريك العوامل الأولية نحو اليسار
Private Sub cmdScrollUP_Click()
' إذا كانت أول لافتة من جهة اليمين ظاهرة تماماً فاخرج
If lblExponent(lblExponent.Count - 1).Left + _
   lblExponent(lblExponent.Count - 1).Width _
         < lblFactor.Width Then Exit Sub
' حرك كل اللافتات نحو اليسار بمقدار ٢٠٠
```

```
Dim i As Integer
For i = 0 To lblX.Count - 1
    lblPrime(i).Left = lblPrime(i).Left - 200
    lblExponent(i).Left = lblExponent(i).Left - 200
    lblX(i).Left = lblX(i).Left - 200
Next
lblPrime(i).Left = lblPrime(i).Left - 200
lblExponent(i).Left = lblExponent(i).Left - 200
End Sub

' عند تحميل النافذة
Private Sub Form_Load()
txtNumber_Change ' استدعاء هذا الحدث لتوليد قواسم العدد ١٢
End Sub

' عند تغيير حجم النافذة
Private Sub Form_Resize()
' إذا تم تصغير النافذة فأخرج
If Me.WindowState = vbMinimized Then Exit Sub
On Local Error Resume Next
' حرك جميع الأدوات على النافذة لتأخذ الموضع و الحجم المناسبين
fraTop.Width = Me.ScaleWidth
picCon.Top = fraTop.Height
picCon.Width = Me.ScaleWidth - picCon.Left * 2
picCon.Height = Me.ScaleHeight - picCon.Top - picCon.Left
txtDSet.Width = fraTop.Width - txtDSet.Left - lblNumber.Left
Err = 0

End Sub

' عند النقر على إحدى اللافتات التي تكون المخطط
Private Sub lbl_MouseDown(Index As Integer, Button As Integer, _
                          Shift As Integer, x As Single, y As Single)
' ربما يكون الهدف من النقر التحريك
mF = True ' أعلم حدث تحرك الفأرة بأنه تم النقر
mX = x  ' خزن إحداثيات نقطة النقر
mY = y
lbl(Index).ZOrder

' إذا كان النقر بالزر الأيمن فالهدف تحديد شبكة جزئية أو مثالي أو مرشحة
If Button = 2 Then
    Dim i As Integer, l As Long, g As Long, n As Long, m As Long
    ' غير ألوان جميع الخطوط إلى اللون الرمادي
    For i = 0 To lin.Count - 1
        lin(i).BorderColor = &HC0C0C0
    Next

    i = Point2
    Point2 = Index
    Point1 = i
    ' عدل عنوان اللافتة ليشير إلى النقطتين المحددتين الجديدتين
    lblS.Caption = "Selected Points: (" & Divisors(Point1).Number _
                   & "," & Divisors(Point2).Number & ")"
    ' إن الحد الأدنى الأعظمي في شبكات القواسم هو القاسم المشترك الأكبر
    g = gcd(Divisors(Point1).Number, Divisors(Point2).Number)
    ' الحد الأعلى الأصغري هو المضاعف المشترك الأصغر
    l = Divisors(Point1).Number * Divisors(Point2).Number / g

    ' كرر من أجل جميع الخطوط
```

```
    For i = 0 To lin.Count - 1
      . n = Divisors(Val(Left(lin(i).Tag, _
                     InStr(lin(i).Tag, ";") - 1))).Number
        m = Divisors(Val(Mid(lin(i).Tag, _
                     InStr(lin(i).Tag, ";") + 1))).Number
        ' هل هذا الخط يشكل جزءاً من الشبكة الجزئية المطلوبة
        If (1 Mod m = 0 And 1 Mod n = 0 And _
            m Mod Divisors(Point1).Number = 0 And _
            n Mod Divisors(Point1).Number = 0) Or _
           (1 Mod m = 0 And 1 Mod n = 0 And _
            m Mod Divisors(Point2).Number = 0 And _
            n Mod Divisors(Point2).Number = 0) Or _
           (m Mod g = 0 And n Mod g = 0 And _
            Divisors(Point1).Number Mod m = 0 And _
            Divisors(Point1).Number Mod n = 0) Or _
           (m Mod g = 0 And n Mod g = 0 And _
            Divisors(Point2).Number Mod m = 0 And _
            Divisors(Point2).Number Mod n = 0) Then
           ' نعم!! غير لونه إلى اللون الأحمر
           lin(i).BorderColor = vbRed
        End If
    Next
End If
End Sub

' عند تحريك الفأرة فوق إحدى اللافتات التي تكون المخطط
Private Sub lbl_MouseMove(Index As Integer, Button As Integer, _
                     Shift As Integer, x As Single, y As Single)
' هل تم النقر على اللافتة
If mF Then
    Dim nX As Integer, nY As Integer
    ' احسب إحداثيات الموضع الجديد
    nX = CInt(ScaleX(x - mX, vbTwips, vbPixels) / (SPA / SPP))
    nY = CInt(ScaleY(y - mY, vbTwips, vbPixels) / (SPA / SPP))
    ' حرك اللافتة إلى الموضع الجديد
    lbl(Index).Move lbl(Index).Left + nX * (SPA / SPP), _
                    lbl(Index).Top + nY * (SPA / SPP)

    Dim i As Integer
    ' حرك جميع الخطوط المتصلة بهذه اللافتة بحيث تصل إلى الموضع الجديد
    For i = 0 To lin.Count - 1
        If Left(lin(i).Tag, Len(Index & ";")) = Index & ";" Then
            lin(i).X1 = lbl(Index).Left + lbl(Index).Width / 2
            lin(i).Y1 = lbl(Index).Top + lbl(Index).Height / 2
        ElseIf Right(lin(i).Tag, Len(";" & Index)) = ";" & Index Then
            lin(i).X2 = lbl(Index).Left + lbl(Index).Width / 2
            lin(i).Y2 = lbl(Index).Top + lbl(Index).Height / 2
        End If
    Next

End If
End Sub
' عند ترك زر الفأرة
Private Sub lbl_MouseUp(Index As Integer, Button As Integer, _
                     Shift As Integer, x As Single, y As Single)
mF = False ' أعلم حدث تحرك الفأرة بأن العملية انتهت
End Sub

' عند النقر المزدوج على اللوحة التي يتم إنشاء المخطط عليها
```

```
Private Sub picCon_DblClick()
' أعد تعيين النقطتين المحددتين '
Point1 = 0
Point2 = 0

lblS.Caption = "Selected Points: (1,1)"

Dim i As Integer
' أعد ألوان كل الخطوط إلى اللون الرمادي '
For i = 0 To lin.Count - 1
    lin(i).BorderColor = &HC0C0C0
Next
End Sub

' عند كبس أحد أزرار لوحة المفاتيح '
Private Sub picCon_KeyUp(KeyCode As Integer, Shift As Integer)
Dim nX As Integer, nY As Integer
' إذا كان الزر المكبوس أحد أزرار الأسهم '
Select Case KeyCode
Case 37 ' اليمين
    nX = 1
    nY = 0
Case 38 ' الأسفل
    nX = 0
    nY = 1
Case 39 ' اليسار
    nX = -1
    nY = 0
Case 40 ' الأعلى
    nX = 0
    nY = -1
End Select
If nX Or nY Then
    'حرك كل عناصر المخطط من خطوط ولافتات إلى الموضع الجديد'
    Dim i As Integer
    For i = 0 To lbl.Count - 1
        lbl(i).Move lbl(i).Left + nX * (SPA / SPP), _
                    lbl(i).Top + nY * (SPA / SPP)
    Next
    For i = 0 To lin.Count - 1
        lin(i).X1 = lin(i).X1 + nX * (SPA / SPP)
        lin(i).X2 = lin(i).X2 + nX * (SPA / SPP)
        lin(i).Y1 = lin(i).Y1 + nY * (SPA / SPP)
        lin(i).Y2 = lin(i).Y2 + nY * (SPA / SPP)
    Next
End If
End Sub

' عند النقر على اللوحة التي يتم إنشاء المخطط عليها '
Private Sub picCon_MouseDown(Button As Integer, Shift As Integer, _
                             x As Single, y As Single)
' إذا كان النقر بالزر الأيمن فأعد تعيين النقطتين المحددتين '
' و ألوان الخطوط كما في إجراء النقر المزدوج '
If Button = 2 Then
    Point1 = 0
    Point2 = 0

    lblS.Caption = "Selected Points: (1,1)"
```

```
    Dim i As Integer

    For i = 0 To lin.Count - 1
        lin(i).BorderColor = &HC0C0C0
    Next
End If
End Sub

' الإجراء الخاص بتوليد قواسم العدد بعد تحسينه ليقبل
' أي عدد من العوامل الأونية
Private Sub GenerateDivisors()
Dim t As Integer, i As Integer, j As Integer, k As Integer

' نقوم بحساب عدد القواسم عن طريق دالة عدد القواسم
t = 1
For i = 0 To UBound(Num.Prime)
    t = t * (Num.Exponent(i) + 1)
Next
' نغير حجم المصفوفة كي تتسع لكل القواسم
ReDim Divisors(t - 1)
' هذه المصفوفة تم بيان وظيفتها سابقا '
ReDim ES(Num.ES)
Lines = 0
' كالعادة نَعُدّ أن كل قاسم هو جداء للعوامل الأولية للعدد الأصلي
' بقوى أصغر أو تساوي مقابلاتها في العدد الأصلي
For i = 0 To t - 1
    ReDim Divisors(i).Prime(UBound(Num.Prime))
    ReDim Divisors(i).Exponent(UBound(Num.Exponent))
    For j = 0 To UBound(Num.Prime)
        Divisors(i).Prime(j) = Num.Prime(j)
    Next
    Divisors(i).Number = 1
Next
ES(0) = 1
' إن أول قاسم هو ١ لذلك نبدأ من القاسم الثاني
For i = 1 To t - 1
    j = 0
    ' نبدأ بزيادة قوى العامل الأولي الأول
    ' فإذا وصلنا إلى أعلى قوة انتقلنا إلى الثاني ... وهكذا '
    Do
        If Divisors(i).Exponent(j) < Num.Exponent(j) Then
            For k = i To t - 1
                Divisors(k).Exponent(j) = Divisors(k).Exponent(j) + 1
            Next
            Exit Do
        Else
            For k = i To t - 1
                Divisors(k).Exponent(j) = 0
            Next
            j = j + 1
        End If
    Loop
    ' نحسب قيمة القاسم عن طريق حساب جداء قوى عوامله الأولية
    For j = 0 To UBound(Num.Prime)
        Divisors(i).Number = Divisors(i).Number * _
                             Divisors(i).Prime(j) ^ _
                             Divisors(i).Exponent(j)
        Divisors(i).ES = Divisors(i).ES + Divisors(i).Exponent(j)
        Lines = Lines + Sgn(Divisors(i).Exponent(j))
```

```
    Next
    ES(Divisors(i).ES) = ES(Divisors(i).ES) + 1
Next
End Sub

' الإجراء الأساسي المحسن الذي يولد المخطط المطلوب
Private Sub GenerateLattice()
Dim Rows As New Collection
Dim i As Integer, j As Integer, k As Integer
Dim l As Integer, Esp() As Integer

' أولا نزيل اللافتات و الخطوط الزائدة أو نحمل المطلوبة بحسب الحاجة
For i = lbl.Count To UBound(Divisors)
    Load lbl(i)
Next
For i = UBound(Divisors) + 1 To lbl.Count - 1
    Unload lbl(i)
Next

For i = lin.Count To Lines - 1
    Load lin(i)
Next
For i = Lines To lin.Count - 1
    Unload lin(i)
Next
' نقوم بإنشاء كائن مجموعة لكل سطر من أسطر المخطط
For i = 0 To Num.ES
    Rows.Add New Collection
Next

ReDim Esp(Num.ES)

' نكرر من أول قاسم إلى آخر قاسم
For i = 0 To UBound(Divisors)
    Dim nX As Integer, nY As Integer
    ' نحسب موضع القاسم اعتمادا على السطر الذي ينتمي إليه
    ' و العناصر التي قبله في السطر نفسه
    nX = Int((picCon.ScaleWidth / 2 + SPA / 2 - _
             ES(Divisors(i).ES) / 2 * SPA + _
             Esp(Divisors(i).ES) * SPA) / (SPA / SPP))
    nY = Int((picCon.ScaleHeight / 2 + Num.ES / 2 * SPA - _
             Divisors(i).ES * SPA) / (SPA / SPP))
    ' نضبط عنوان اللافتة و موضعها
    lbl(i).Caption = Trim(Divisors(i).Number)
    lbl(i).Tag = Trim(Divisors(i).Number)
    lbl(i).Move nX * (SPA / SPP) - lbl(i).Width / 2, _
               nY * (SPA / SPP) - lbl(i).Height / 2
    Esp(Divisors(i).ES) = Esp(Divisors(i).ES) + 1
    lbl(i).Visible = True
    lbl(i).ZOrder
    ' نضيف رقم القاسم إلى كائن السطر الذي ينتمي إليه
    Rows(Divisors(i).ES + 1).Add i
Next

' نبدأ الآن بتحريك الخطوط بين الأسطر لذلك نبدأ من السطر الأول
For i = 1 To Rows.Count - 1
    ' نصل كل عنصر من السطر
    For j = 1 To Rows(i).Count
        ' بالعناصر التي يتسمها من السطر الذي يعلوه
```

```
        For k = 1 To Rows(i + 1).Count
            Dim xi As Integer, yi As Integer
            xi = Rows(i)(j)
            yi = Rows(i + 1)(k)
            ' هل علاقة يقسم محققة
            If Divisors(yi).Number Mod Divisors(xi).Number = 0 Then
                ' حرك أحد الخطوط ليصل بين العنصرين
                lin(l).X1 = lbl(xi).Left + lbl(xi).Width / 2
                lin(l).Y1 = lbl(xi).Top + lbl(xi).Height / 2
                lin(l).X2 = lbl(yi).Left + lbl(yi).Width / 2
                lin(l).Y2 = lbl(yi).Top + lbl(yi).Height / 2
                lin(l).Visible = True
                lin(l).Tag = xi & ";" & yi
                l = l + 1
            End If
        Next
    Next
Next

End Sub

' الإجراء الذي يقوم بحساب القاسم المشترك الأكبر لعددين بوساطة خوارزمية
' إقليدس الشهيرة و لا حاجة لشرحها
Private Function gcd(ByVal a As Long, ByVal b As Long) As Long
Dim q As Long, r As Long

gcd = b
Do
     q = a \ b
     r = a Mod b
     If r <> 0 Then gcd = r
     a = b
     b = r
Loop Until r = 0
End Function

' عند تغيير النص في مربع النص المخصص لإدخال العدد
Private Sub txtNumber_Change()
Dim i As Integer, c As Integer

If Val(txtNumber.Text) > 2147483647 Then
     txtNumber.Text = "2147483647"
     Exit Sub
End If
If Val(txtNumber.Text) <= 0 Then
     txtNumber.Text = "1"
     Exit Sub
End If

Num.Number = Val(txtNumber.Text)
' حلل العدد إلى عوامله الأولية
c = GetFactors(Num)

' نظهر العوامل الأولية و فوقها قواها و بينها إشارات الضرب
' عن طريق اللافتات المخصصة لذلك
lblX(0).Visible = c
For i = lblPrime.Count To c
     Load lblPrime(i)
     Load lblExponent(i)
```

```
    lblPrime(i).Visible = True
    lblExponent(i).Visible = True
Next
For i = lblX.Count To c - 1
    Load lblX(i)
    lblX(i).Visible = True
Next
For i = c + 1 To lblPrime.Count - 1
    Unload lblPrime(i)
    Unload lblExponent(i)
Next
For i = c To lblX.Count - 1
    If i Then Unload lblX(i)
Next
For i = 0 To c
    lblPrime(i).Caption = Num.Prime(i)
    If Num.Exponent(i) <= 1 Then
        lblExponent(i).Caption = ""
    Else
        lblExponent(i).Caption = Num.Exponent(i)
    End If
    If i Then
        lblX(i - 1).Left = lblExponent(i - 1).Left + _
                           lblExponent(i - 1).Width + 45
        lblPrime(i).Left = lblX(i - 1).Left + lblX(i - 1).Width + 45
        lblExponent(i).Left = lblPrime(i).Left + lblPrime(i).Width
    Else
        lblPrime(0).Left = 90
        lblExponent(0).Left = lblPrime(0).Left + lblPrime(0).Width
    End If
Next
For i = i To lblPrime.Count - 1
    Unload lblPrime(i)
Next
' نستدعي إجراء توليد القواسم
GenerateDivisors

Dim s As String
' نولد العبارة التي تمثل مجموعة القواسم
s = ""
For i = 0 To UBound(Divisors)
    s = s & Divisors(i).Number & ", "
Next

s = Mid(s, 1, Len(s) - 2)
' نضع العبارة التي تمثل مجموعة القواسم في مربع النص المخصص لذلك
txtDSet.Text = "D(" & Num.Number & ")={" & s & "}"
End Sub
' الإجراء الخاص بتحليل العدد إلى عوامله الأولية
' يعتمد على الخوارزمية التي أسلفنا ذكرها
Private Function GetFactors(Number As Primes) As Integer
Dim d As Long, r As Long, n As Integer

n = -1
r = Number.Number
Number.ES = 0

ReDim Number.Prime(0)
ReDim Number.Exponent(0)
```

```
While r Mod 2 = 0
    n = 0
    Number.Prime(0) = 2
    Number.Exponent(0) = Number.Exponent(0) + 1
    r = r / 2
    Number.ES = Number.ES + 1
Wend

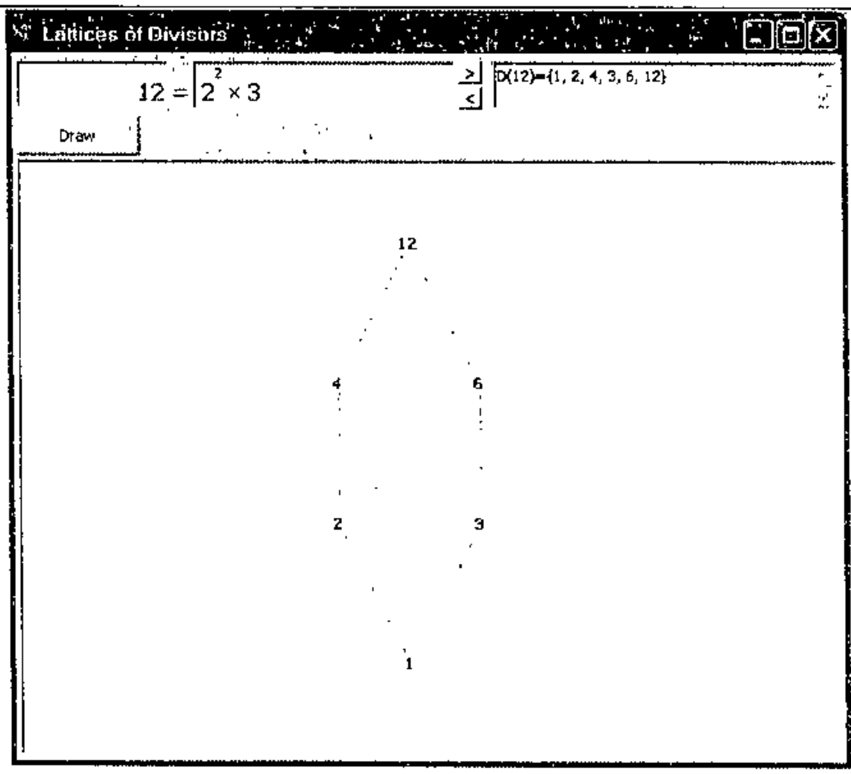
d = 3
While d <= Sqr(r)
    While r Mod d = 0
        If Number.Prime(UBound(Number.Prime)) = d Then
            Number.Exponent(n) = Number.Exponent(n) + 1
        Else
            n = n + 1
            ReDim Preserve Number.Prime(n)
            ReDim Preserve Number.Exponent(n)
            Number.Prime(n) = d
            Number.Exponent(n) = 1
        End If
        r = r / d
        Number.ES = Number.ES + 1
    Wend
    d = d + 2
Wend

If n = -1 Then
    n = 0
    ReDim Preserve Number.Prime(0)
    ReDim Preserve Number.Exponent(0)
    Number.Prime(0) = Number.Number
    Number.Exponent(0) = IIf(Number.Number = 1, 0, 1)
    Number.ES = 1
Else
    If r <> 1 Then
        If Number.Prime(n) = r Then
            Number.Exponent(n) = Number.Exponent(n) + 1
        Else
            n = n + 1
            ReDim Preserve Number.Prime(n)
            ReDim Preserve Number.Exponent(n)
            Number.Prime(n) = r
            Number.Exponent(n) = 1
        End If
        Number.ES = Number.ES + 1
    End If
End If

GetFactors = n
End Function
```

إن الإجراء GetFactors السابق يستعمل الخوارزمية التي أوردناها لتحليل العدد إلى عوامله الأولية، و بدلاً من حساب المتغير Num الذي يمثل العدد وعوامله الأولية و الـذي حسبناه سابقاً ضمن الحدث Change لمربعات النصوص العشرة المخصصة لإدخال العوامل الأولية

و قواها فإننا نستدعى هذه الإجــراء ضــمن الحــدث Change الخــاص بمربــع الــنص txtNumber المخصص لإدخال العدد فيه، ثم نولد مصفوفة من اللافتات و نرتبهـا بحيـث تعطي الشكل النظامي لكيفية كتابة العدد كجداء لقوى عوامل أولية. ثـم نـستدعي الإجـراء GenerateDivisors المعدل وهو يولد جميع قواسم العدد مهما كان عدد العوامـل الأوليـة وقواها، فبدلاً من استخدام خمسة تكرارات متداخلة نحسب دالة عدد القواسم، بعد ذلك نوسـع المصفوفة Divisors ليصبح عدد عناصرها يساوي عدد القواسم، و بما أننا نعدُّ كل قاسم هو جداء العوامل الأولية للعدد المعطى و لكنها مرفوعة إلى قوى تتراوح بين الصفر و أكبر قوة للعامل الأولي تقسم العدد المعطى فإننا نقوم بتكرار ونعطي كل عامل أولي قوى متزايدة حتى نحصل على جميع القواسم.



الشكل ٧: واجهة البرنامج المحسن

يشمل التحسين أيضاً الإجراء GenerateLattice الذي يختلف عن سابقه فـي أنـه يـضع العنصر مباشرة في موقعه الصحيح مادام السطرُ الذي يقع فيه العنصر معروفاً كما أوضـحنا في ٢§١. وحللنا مسألة وصل العناصر عن طريق استخدام الكـائن Rows مـن الـصنف Collection [7] الذي يعمل كمصفوفة ديناميكية يمكن إضافة أو إزالة أي مـن عناصـرها مباشرة، و نستعمل الكائن Rows هنا لتخزين عناصر كل سطر من الأسطر، وبالتالي تنعدم

الحاجة السابقة لترتيب عناصر السطر الأول ثم عناصر السطر الثاني ثم الوصل بين السطرين ثم عناصر السطر الثالث ثم الوصل بينه و بين السطر الثاني ...الخ، فبهذا التحسين نضع جميع العناصر في أماكنها في تكرار واحد ثم نصل بينها في تكرار آخر ونعرف عناصر كل سطر عن طريق الكائن Rows.

أما في الأحداث MouseDown و MouseMove و MouseUp فوضعنا الشيفرة البرمجية اللازمة لتحريك عناصر المخطط بسحبها و إفلاتها بالإضافة إلى إمكانية تلوين أي شبكة جزئية من المخطط عن طريق تحديد أي عنصرين من الشبكة الجزئية بالنقر بالزر الأيمن عليهما بشكل متتالٍ.

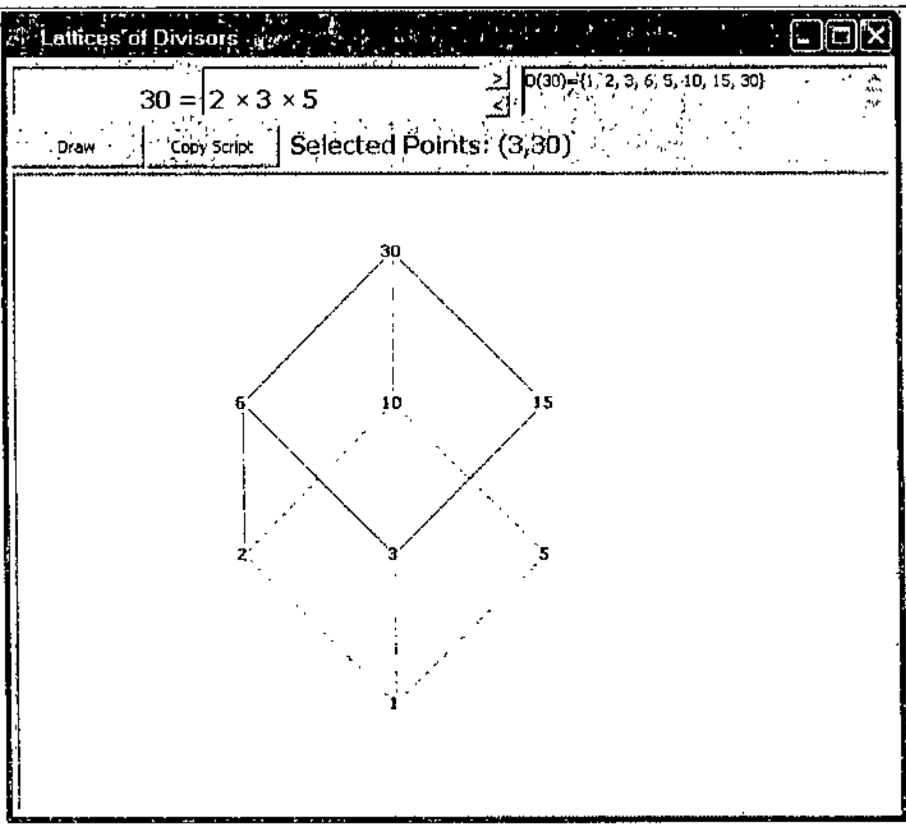
٥. تنفيذ البرنامج و نتائج الاستخدام Executing and Results

نورد الآن بعض الأمثلة عن شبكات طبقناها على البرنامج السابق:

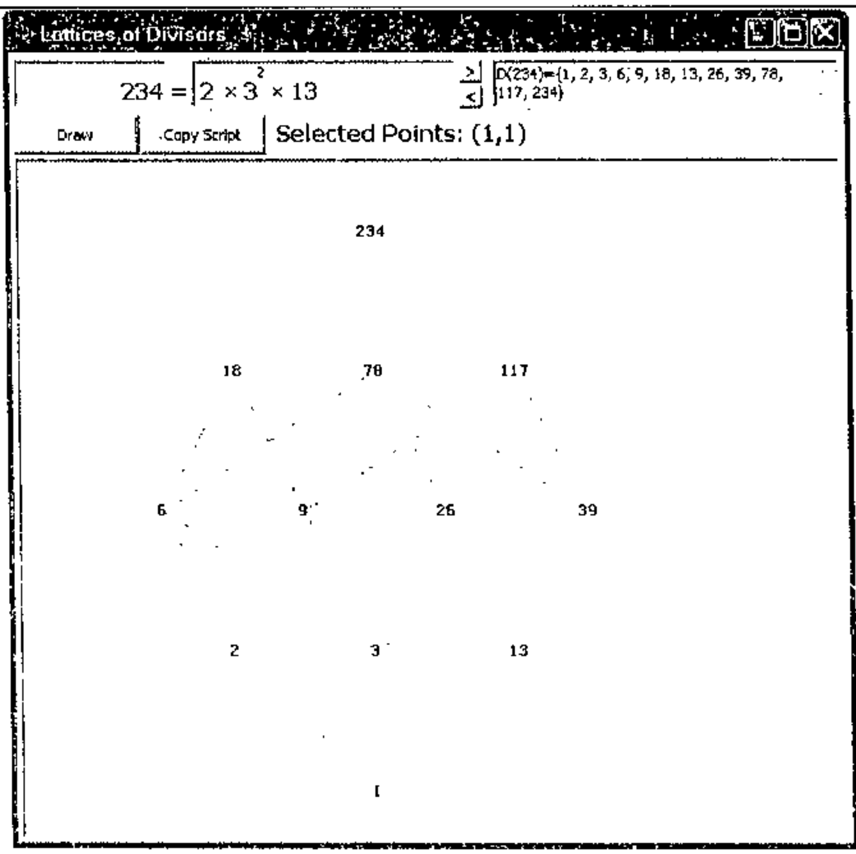
٥-١ مثال. عند كتابة العدد 30 في مربع النص الخاص بإدخال العدد، فيتم تلقائياً تحليل العدد إلى عوامله الأولية و توليد مجموعة قواسمه، و بالنقر على الزر Draw يظهر مخطط هاس للشبكة (D(30), |) ، و لتحديد المرشحة [3) مثلاً ننقر بالزر الأيمن على 3 ثم ننقر بالزر الأيمن على العنصر الأكبر في الشبكة أي على 30 فيتم تلوين المرشحة المطلوبة باللون الأحمر كما يظهر في الشكل ٨.

٥-٢ مثال. عند كتابة العدد 234 في مربع النص الخاص بإدخال العدد فيتم تلقائياً تحليل العدد إلى عوامله الأولية و توليد مجموعة قواسمه، و بالنقر على الزر Draw يظهر مخطط هاس للشبكة (D(234), |).

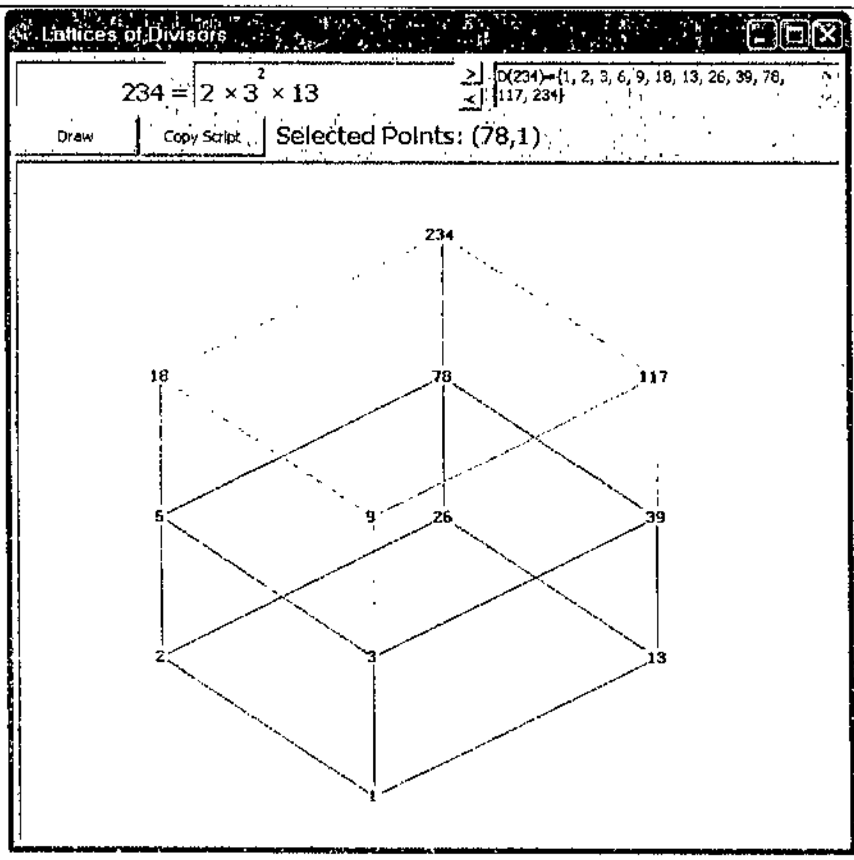
و لتحسين المخطط الناتج نحرك العناصر بالسحب و الإفلات بحيث نحصل على الشكل المناسب، ثم لتحديد أحد المثاليات و ليكن المثالي [78) ننقر بالزر الأيمن على 78 ثم ننقر بالزر الأيمن على العنصر الأصغر في الشبكة أي على 1 فيتم تلوين المثالي المطلوب باللون الأحمر.



الشكل ٨: الشبكة (|, D(30)) و ضمنها المرشحة (3]



الشكل ٩: الشبكة (| , D(234)) قبل تحسين المخطط



الشكل ١٠: الشبكة السابقة بعد تحسينها بسحب العناصر و تحديد المثالي [78)

§٣ الفصل الثالث

إنشاء مخطط هاس لشبكات المجموعات الجزئية

Creating Hasse Diagram of Subsets Lattices

١. التعبير عن علاقة الاحتواء برمجياً Programming `IN` Relation

هدفنا في هذا الفصل إيجاد تطبيق برمجي لإنشاء شبكة المجموعات الجزئية في مجموعة منتهية معطاة. و تعدُّ هذه الشبكة من أكثر أنواع الشبكات التي تمت دراستها و لها أهمية بالغة لما يربطها بجبر المجموعات و التبولوجيا و غيرها من فروع الرياضيات.

إن علاقة الاحتواء ليست مُعرّفة بشكل مباشر ضمن العمليات التي بمقدور الكمبيوتر القيام بها، لذا علينا تعريف علاقة الاحتواء بحيث يجريها الكمبيوتر بشكل مباشر لا يزيد من تعقيد البرنامج.

من أجل ذلك وضعنا تقابلاً بين علاقة الاحتواء و عملية and المنطقية على النحو التالي:
لتكن $E = \{x, y, z, v, w\}$ هي المجموعة التي نريد توليد شبكة المجمعات الجزئية فيها، ولتكن $A = \{x\}$ و $B = \{x, y\}$ عنصرين من الشبكة، واضح أن $A \subseteq B$ لكن بالنسبة للكمبيوتر يتطلب البرهان على ذلك أن نرقم عناصر المجموعة E من الصفر و حتى 4 ، ثم نمثل كل مجموعة بمصفوفة من البتات بحيث إذا كان العنصر ذو الرقم i ينتمي إلى المجموعة الجزئية فإننا نضع 1 في البت المقابل لهذا العنصر و إلا فإننا نضع 0 و يكون تمثيل المجموعات السابقة وفق هذا المفهوم على النحو التالي:

E				
1	1	1	1	1

A				
1	0	0	0	0

B				
1	1	0	0	0

و يمكن أن نضع تعريفاً لعلاقة الاحتواء على النحو التالي:

A				
1	0	0	0	0

And

B				
1	1	0	0	0

=

A				
1	0	0	0	0

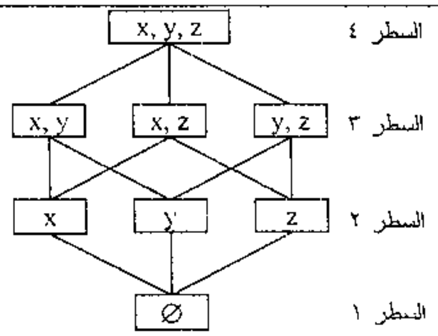
أي الشرط اللازم و الكافي ليكون $A \subseteq B$ هو أن يكون A And B = A.

٢. إنشاء مخطط الشبكة Creating the Diagram

عند دراسة أي من شبكات المجموعات الجزئية في مجموعة منتهية و برسم مخطط هاس الذي يمثلها، نرى أمرين أساسيين يميزان هذا النوع من الشبكات:

- إن عناصر شبكة المجموعات الجزئية تتوضع على أسطر عددها يساوي عدد عناصر المجموعة الأساسية زائداً واحداً، بحيث إذا رقمنا الأسطر من الأسفل إلى الأعلى، فإن كل سطر يحوي المجموعات الجزئية التي عدد عناصرها يساوي رقم السطر ناقصاً واحداً.
- إن عدد الخطوط التي تصل بين مجموعة من سطر ما و بين مجموعات السطر الذي تحته يساوي عدد عناصر هذه المجموعة.

كمثال بسيط على ذلك لنأخذ الشبكة $(P(\{x, y, z\}), \subseteq)$ كما في (الشكل ١١).



الشكل ١١: المخطط التفصيلي للشبكة $(P(\{x, y, z\}), \subseteq)$

نلاحظ أن السطر 1 يحوي المجموعة الخالية التي عدد عناصرها صفر، والسطر 2 الذي يعلوه يحوي المجموعات التي عدد عناصرها يساوي 1؛ وهي $\{x\}$ و $\{y\}$ و $\{z\}$. أما السطر 3 فيحوي المجموعات التي عدد عناصرها يساوي 2 وهي $\{x, y\}$ و $\{x, z\}$ و $\{y, z\}$. وهكذا حتى الوصول إلى السطر الأخير فنجد فيه المجموعة التي عدد عناصرها يساوي عدد عناصر المجموعة الأساسية، أي يحوي فقط المجموعة $\{x, y, z\}$.

ومن جهة أخرى، فإن عدد عناصر المجموعة $\{x, y, z\}$ هو 3 لذلك ترتبط بالسطر الذي تحتها بثلاثة خطوط، بينما عدد عناصر المجموعات $\{x, y\}$ و $\{x, z\}$ و $\{y, z\}$ هو 2 لهذا

يرتبط كل منهما بالسطر الذي تحتها بخطين، بينما ترتبط المجموعات {x} و {y} و {z} بالسطر الذي تحتها بخط واحد لأنها وحيدة العنصر.

و معلوم أنه بفرض عدد عناصر مجموعة منتهية E هو n يكون عدد عناصر أسرة المجموعات P(E) هو 2^n. لذلك فإن عدد عناصر شبكة المجموعات الجزئية يمكن حسابه بسهولة، لكن الأمر المهم هو كيفية توليد عناصر هذه الشبكة.

بالعودة إلى تمثيل المجموعات كمصفوفة من البتات نجد أن المجموعة الأساسية -ولنفرض أنها مكونة من ثلاثة عناصر {x, y, z} مثلاً- تتمثل على الشكل 111 لأنها تحوي العناصر الثلاثة، و يكون عدد المجموعات الجزئية فيها هو $2^3 = 8$، الآن إذا كتبنا الأعداد من صفر وحتى 7 بالنظام الثنائي نجد أنها على النحو:

$0 = (000)_2 \Leftrightarrow P_0 = \varnothing$
$1 = (100)_2 \Leftrightarrow P_1 = \{x\}$
$2 = (010)_2 \Leftrightarrow P_2 = \{y\}$
$3 = (110)_2 \Leftrightarrow P_3 = \{x, y\}$
$4 = (001)_2 \Leftrightarrow P_4 = \{z\}$
$5 = (101)_2 \Leftrightarrow P_5 = \{x, z\}$
$6 = (011)_2 \Leftrightarrow P_6 = \{y, z\}$
$7 = (111)_2 \Leftrightarrow P_7 = \{x, y, z\}$

و نلاحظ أن كلاً من هذه الأعداد الثمانية يكافئ أحد عناصر شبكة المجموعات الجزئية المطلوبة.

٣. كتابة الشيفرة البرمجية Typing the Code

وعلى ضوء ما سبق يمكننا الآن إنشاء برنامج مشابه للبرنامج الذي قدمناه في الفصل الثاني وهذا البرنامج مختص بإنشاء شبكة المجموعات الجزئية لمجموعة منتهية معطاة. تكمن التغيرات الأساسية في الإجراء GenerateSubsets الذي يولد المجموعات الجزئية بالطريقة السابقة. بالإضافة إلى تحديده العناصر التي تنتمي إلى مجموعة بتطبيق التكافؤ التالي:

العنصر ذو الرقم i ينتمي إلى المجموعة الجزئية n $\Leftrightarrow$ n and $2^i \neq 0$

حيث تقوم العملية and باختبار وجود 1 في البت المقابل للعنصر ذو الرقم i.

وفيما يلي نورد الشيفرة البرمجية الكاملة للنموذج frmMain الذي يمثل البرنـــامج الخـــاص بإنشاء شبكات المجموعات الجزئية و يسمح بتحديد شبكات جزئية، و تحريك عناصر المخطط بشكل مشابه لبرنامج إنشاء شبكات القواسم.

الشيفرة البرمجية الكاملة للنموذج frmMain

```
VERSION 5.00
Begin VB.Form frmMain
   Caption         =   "Lattices of Subsets"
   ClientHeight    =   7590
   ClientLeft      =   60
   ClientTop       =   450
   ClientWidth     =   8880
   BeginProperty Font
      Name            =   "Tahoma"
      Size            =   8.25
      Charset         =   178
      Weight          =   400
      Underline       =   0   'False
      Italic          =   0   'False
      Strikethrough   =   0   'False
   EndProperty
   Icon            =   "Main.frx":0000
   LinkTopic       =   "Form1"
   ScaleHeight     =   7590
   ScaleWidth      =   8880
   StartUpPosition =   2  'CenterScreen
   Begin VB.Frame fraTop
      BorderStyle     =   0  'None
      ForeColor       =   &H00000000&
      Height          =   1140
      Left            =   0
      TabIndex        =   2
      Top             =   0
      Width           =   8115
      Begin VB.TextBox txtElements
         BeginProperty Font
            Name            =   "Tahoma"
            Size            =   12
            Charset         =   178
            Weight          =   700
            Underline       =   0   'False
            Italic          =   0   'False
            Strikethrough   =   0   'False
         EndProperty
         Height          =   405
         Left            =   1350
         TabIndex        =   8
         Text            =   "a,b,c"
         Top             =   135
         Width           =   3645
      End
      Begin VB.CommandButton cmdCopyScript
         Caption         =   "Copy Script"
         Height          =   420
         Left            =   1395
         TabIndex        =   7
```

```
      Top              =    675
      Width            =    1320
   End
   Begin VB.CommandButton cmdDraw
      Caption          =    "Draw"
      Height           =    420
      Left             =    45
      TabIndex         =    6
      Top              =    675
      Width            =    1320
   End
   Begin VB.TextBox txtPSet
      BackColor        =    &H8000000F&
      Height           =    510
      Left             =    5130
      Locked           =    -1  'True
      MultiLine        =    -1  'True
      ScrollBars       =    2  'Vertical
      TabIndex         =    4
      Text             =    "Main.frx":030A
      Top              =    90
      Width            =    2625
   End
   Begin VB.Label Label1
      AutoSize         =    -1  'True
      BackStyle        =    0  'Transparent
      Caption          =    "Elements:"
      BeginProperty Font
         Name             =    "Tahoma"
         Size             =    12
         Charset          =    178
         Weight           =    700
         Underline        =    0   'False
         Italic           =    0   'False
         Strikethrough    =    0   'False
      EndProperty
      ForeColor        =    &H00000000&
      Height           =    285
      Left             =    90
      TabIndex         =    5
      Top              =    180
      Width            =    1215
   End
   Begin VB.Label lblS
      AutoSize         =    -1  'True
      Caption          =    "Selected Points: ({},{})"
      BeginProperty Font
         Name             =    "Tahoma"
         Size             =    12
         Charset          =    178
         Weight           =    700
         Underline        =    0   'False
         Italic           =    0   'False
         Strikethrough    =    0   'False
      EndProperty
      Height           =    285
      Left             =    2835
      TabIndex         =    3
      Top              =    720
```

```
            Width              =   2940
         End
      End
      Begin VB.PictureBox picCon
         BackColor        =   &H00FFFFFF&
         Height           =   4470
         Left             =   45
         ScaleHeight      =   294
         ScaleMode        =   3  'Pixel
         ScaleWidth       =   429
         TabIndex         =   0
         Top              =   1170
         Width            =   6495
         Begin VB.Label lbl
            Alignment        =   2  'Center
            AutoSize         =   -1  'True
            BackColor        =   &H00FFFFFF&
            Caption          =   "#"
            BeginProperty Font
               Name             =   "Tahoma"
               Size             =   8.25
               Charset          =   178
               Weight           =   700
               Underline        =   0   'False
               Italic           =   0   'False
               Strikethrough    =   0   'False
            EndProperty
            ForeColor        =   &H00000000&
            Height           =   195
            Index            =   0
            Left             =   2295
            TabIndex         =   1
            Top              =   3105
            Visible          =   0   'False
            Width            =   150
         End
         Begin VB.Line lin
            BorderColor      =   &H00C0C0C0&
            Index            =   0
            Visible          =   0   'False
            X1               =   117
            X2               =   120
            Y1               =   216
            Y2               =   171
         End
      End
   End
Attribute VB_Name = "frmMain"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit

' هذه المتغيرات مطلوبة لتحريك عناصر المخطط
Dim mF As Boolean
Dim mX As Single
Dim mY As Single
```

الشيفرة البرمجية الكاملة للنموذج frmMain

```
' النقطتان اللتان تم تحديدهما لتلوين الشبكة الجزئية المولدة بهما
Dim Point1 As Integer
Dim Point2 As Integer

Const SPA = 100
Const SPP = 2

' قوى العدد ٢ نحتاجها لتحديد انتماء العناصر إلى المجموعات الجزئية
Dim PowersOf2(30) As Long

Dim Num As Integer          'عدد العناصر في المجموعة الأساسية
Dim Elements() As String    'مصفوفة تمثل عناصر المجموعة الأساسية
Dim mCount As Long          'عدد عناصر المخطط
Dim Subsets() As String     'مصفوفة تمثل المجموعات الجزئية
Dim sCount() As Integer     'عدد عناصر كل مجموعة من المجموعات الجزئية
Dim ES() As Integer         'مشابه للبرنامج الخاص بشبكات القواسم
Dim Lines As Long           'عدد الخطوط المطلوبة لرسم المخطط

' عند النقر على زر الرسم
Private Sub cmdDraw_Click()
GenerateLattice ' استدعاء إجراء توليد الشبكة
picCon.SetFocus
End Sub

' عند تحميل النافذة
Private Sub Form_Load()
Dim i As Long
' نولد قوى العدد ٢ لحاجتنا إليها لاحقا و للاستغناء عن حسابها كل مرة
For i = 0 To 30
    PowersOf2(i) = 2 ^ i
Next i
txtElements_Change
End Sub

' عند تغيير حجم النافذة
Private Sub Form_Resize()
' إذا تم تصغير النافذة فأخرج
If Me.WindowState = vbMinimized Then Exit Sub
On Local Error Resume Next
' حرك جميع الأدوات على النافذة لتأخذ الموضع و الحجم المناسبين
fraTop.Width = Me.ScaleWidth
picCon.Top = fraTop.Height
picCon.Width = Me.ScaleWidth - picCon.Left * 2
picCon.Height = Me.ScaleHeight - picCon.Top - picCon.Left
txtPSet.Width = fraTop.Width - txtPSet.Left - Label1.Left
Err = 0
End Sub

' عند النقر على إحدى اللافتات التي تكون المخطط
Private Sub lbl_MouseDown(Index As Integer, Button As Integer, _
                         Shift As Integer, x As Single, Y As Single)
' ربما يكون الهدف من النقر التحريك
mF = True ' أعلم الحدث تحرك الفأرة بأنه تم النقر
mX = x  ' خزن إحداثيات نقطة النقر
mY = y
lbl(Index).ZOrder

' إذا كان النقر بالزر الأيمن فالهدف تحديد شبكة جزئية أو مثالي أو مرشحة
If Button = 2 Then
```

```
    Dim i As Integer, l As Long, g As Long, n As Long, m As Long

    i = Point2
    Point2 = Index
    Point1 = i

    ' عدل عنوان اللافتة ليشير إلى النقطتين المحددتين الجديدتين
    lblS.Caption = "Selected Points: ({" & Subsets(Point1) & "},{" _
                   & Subsets(Point2) & "})"

    ' إن الحد الأدنى الأعظمي في شبكات المجموعات هو التقاطع
    g = Point1 And Point2
    ' الحد الأعلى الأصغري هو الاجتماع
    l = Point1 Or Point2

    ' كرر من أجل جميع الخطوط
    For i = 0 To lin.Count - 1
        n = Val(Left(lin(i).Tag, InStr(lin(i).Tag, ";") - 1))
        m = Val(Mid(lin(i).Tag, InStr(lin(i).Tag, ";") + 1))

        ' هل هذا الخط يشكل جزءاً من الشبكة الجزئية المطلوبة
        If ((l And m) = m And (l And n) = n And _
           (m And Point1) = Point1 And (n And Point1) = Point1) Or _
          ((l And m) = m And (l And n) = n And _
           (m And Point2) = Point2 And (n And Point2) = Point2) Or _
          ((m And g) = g And (n And g) = g And _
           (Point1 And m) = m And (Point1 And n) = n) Or _
          ((m And g) = g And (n And g) = g And (Point2 And m) = m _
           And (Point2 And n) = n) Then
            ' نعم!! غير لونه إلى اللون الأحمر
            lin(i).BorderColor = vbRed
        Else
            ' لا! غير لونه إلى اللون الرمادي
            lin(i).BorderColor = &HC0C0C0
        End If
    Next
End If
End Sub

' عند تحريك الفأرة فوق إحدى اللافتات التي تكون المخطط
Private Sub lbl_MouseMove(Index As Integer, Button As Integer, _
                          Shift As Integer, x As Single, Y As Single)
' هل تم النقر على اللافتة
If mF Then
    Dim nX As Integer, nY As Integer
    ' احسب إحداثيات الموضع الجديد
    nX = CInt(ScaleX(x - mX, vbTwips, vbPixels) / (SPA / SPP))
    nY = CInt(ScaleY(Y - mY, vbTwips, vbPixels) / (SPA / SPP))
    ' حرك اللافتة إلى الموضع الجديد
    lbl(Index).Move lbl(Index).Left + nX * (SPA / SPP), _
                    lbl(Index).Top + nY * (SPA / SPP)

    Dim i As Integer
    ' حرك جميع الخطوط المتصلة بهذه اللافتة بحيث تصل إلى الموضع الجديد
    For i = 0 To lin.Count - 1
        If Left(lin(i).Tag, Len(Index & ";")) = Index & ";" Then
            lin(i).X1 = lbl(Index).Left + lbl(Index).Width / 2
            lin(i).Y1 = lbl(Index).Top + lbl(Index).Height / 2
        ElseIf Right(lin(i).Tag, Len(";" & Index)) = ";" & Index Then
```

```
            lin(i).X2 = lbl(Index).Left + lbl(Index).Width / 2
            lin(i).Y2 = lbl(Index).Top + lbl(Index).Height / 2
        End If
    Next

End If
End Sub

' عند ترك زر الفأرة
Private Sub lbl_MouseUp(Index As Integer, Button As Integer, _
                        Shift As Integer, x As Single, Y As Single)
mF = False ' أعلم حدث تحرك الفأرة بأن العملية انتهت
End Sub

' عند النقر المزدوج على اللوحة التي يتم إنشاء المخطط عليها
Private Sub picCon_DblClick()
' أعد تعيين النقطتين المحددتين
Point1 = 0
Point2 = 0

lblS.Caption = "Selected Points: ({},{})"

Dim i As Integer
' أعد ألوان كل الخطوط إلى اللون الرمادي
For i = 0 To lin.Count - 1
    lin(i).BorderColor = &HC0C0C0
Next
End Sub

' عند كبس أحد أزرار لوحة المفاتيح
Private Sub picCon_KeyUp(KeyCode As Integer, Shift As Integer)
Dim nX As Integer, nY As Integer
' إذا كان الزر المكبوس أحد أزرار الأسهم
Select Case KeyCode
Case 37 ' اليمين
    nX = 1
    nY = 0
Case 38 ' الأسفل
    nX = 0
    nY = 1
Case 39 ' اليسار
    nX = -1
    nY = 0
Case 40 ' الأعلى
    nX = 0
    nY = -1
End Select
If nX Or nY Then
    Dim i As Integer
    ' حرك كل عناصر المخطط من خطوط ولافتات إلى الموضع الجديد
    For i = 0 To lbl.Count - 1
        lbl(i).Move lbl(i).Left + nX * (SPA / SPP), _
                    lbl(i).Top + nY * (SPA / SPP)
    Next
    For i = 0 To lin.Count - 1
        lin(i).X1 = lin(i).X1 + nX * (SPA / SPP)
        lin(i).X2 = lin(i).X2 + nX * (SPA / SPP)
        lin(i).Y1 = lin(i).Y1 + nY * (SPA / SPP)
        lin(i).Y2 = lin(i).Y2 + nY * (SPA / SPP)
```

```
    Next
End If
End Sub

' عند النقر على اللوحة التي يتم إنشاء المخطط عليها '
Private Sub picCon_MouseDown(Button As Integer, Shift As Integer, _
                              x As Single, Y As Single)
' إذا كان النقر بالزر الأيمن فأعد تعيين النقطتين المحددتين '
' و ألوان الخطوط كما في إجراء النقر المزدوج '
If Button = 2 Then
    Point1 = 0
    Point2 = 0

    lblS.Caption = "Selected Points: ({},{})"

    Dim i As Integer

    For i = 0 To lin.Count - 1
        lin(i).BorderColor = &HC0C0C0
    Next
End If
End Sub

' الإجراء الخاص بتوليد المجموعات الجزئية '
Private Sub GenerateSubsets()
Dim s As String, i As Long, j As Integer

s = ""
Lines = 0
ReDim ES(Num)
mCount = 2 ^ Num  ' نحسب عدد المجموعات الجزئية '

ReDim Subsets(mCount - 1)  ' نضبط حجم المصفوفتين ليتسع لكل المجموعات
ReDim sCount(mCount - 1)

' نكرر من الصفر و حتى عدد المجموعات الجزئية ناقصاً واحداً '
For i = 0 To mCount - 1
    s = ""
    ' لنحديد العناصر التي تنتمي إلى هذه المجموعة '
    For j = 0 To Num - 1
        ' نطبق الاختبار الذي أوردناه سابقاً '
        If i And PowersOf2(j) Then
            s = s & Elements(j) & ", "
            sCount(i) = sCount(i) + 1
        End If
    Next
    If sCount(i) Then Subsets(i) = Left(s, Len(s) - 2)
    ' نضيف إلى عدد الخطوط عدد عناصر هذه المجموعة '
    Lines = Lines + sCount(i)
    ' نضيف واحداً إلى عدد العناصر التي تنتمي إلى هذا السطر '
    ES(sCount(i)) = ES(sCount(i)) + 1
Next
End Sub

' الإجراء الأساسي الحسن الذي يولد المخطط المطلوب '
Private Sub GenerateLattice()
Dim Rows As New Collection
Dim i As Long, j As Integer, k As Integer
Dim l As Integer, Esp() As Integer
```

```
' أولا نزيل اللافتات و الخطوط الزائدة أو نحمل المطلوبة بحسب الحاجة
For i = lbl.Count To mCount - 1
    Load lbl(i)
Next
For i = mCount To lbl.Count - 1
    Unload lbl(i)
Next

For i = lin.Count To Lines - 1
    Load lin(i)
Next
For i = Lines To lin.Count - 1
    Unload lin(i)
Next
' نقوم بإنشاء كائن مجموعة لكل سطر من أسطر المخطط
For i = 0 To Num
    Rows.Add New Collection
Next

ReDim Esp(Num)

' نكرر من أول مجموعة جزئية إلى آخر مجموعة جزئية
For i = 0 To mCount - 1
    Dim nX As Integer, nY As Integer
    ' نحسب موضع القاسم اعتمادا على السطر الذي ينتمي إليه
    ' و العناصر التي قبله في السطر نفسه
    nX = Int((picCon.ScaleWidth / 2 + SPA / 2 - _
             ES(sCount(i)) / 2 * SPA + _
             Esp(sCount(i)) * SPA) / (SPA / SPP))
    nY = Int((picCon.ScaleHeight / 2 + Num / 2 * SPA - _
             sCount(i) * SPA) / (SPA / SPP))
    ' نضبط عنوان اللافتة و موضعها
    lbl(i).Tag = Subsets(i)
    lbl(i).Caption = "{" & Subsets(i) & "}"
    lbl(i).Move nX * (SPA / SPP) - lbl(i).Width / 2, _
               nY * (SPA / SPP) - lbl(i).Height / 2
    Esp(sCount(i)) = Esp(sCount(i)) + 1
    lbl(i).Visible = True
    lbl(i).ZOrder
    ' نضيف رقم المجموعة إلى كائن السطر الذي تنتمي إليه
    Rows(sCount(i) + 1).Add i
Next

' نبدأ الآن بتحريك الخطوط بين الأسطر لذلك نبدأ من السطر الأول
For i = 1 To Rows.Count - 1
    ' نصل كل عنصر من السطر
    For j = 1 To Rows(i).Count
        ' بالعناصر التي يقسمها من السطر الذي يعلوه
        For k = 1 To Rows(i + 1).Count
            Dim xi As Long, yi As Long
            xi = Rows(i)(j)
            yi = Rows(i + 1)(k)
            ' هل علاقة الاحتواء محققة
            If (xi And yi) = xi Then
                ' حرك أحد الخطوط ليصل بين المجموعتين
                lin(l).X1 = lbl(xi).Left + lbl(xi).Width / 2
                lin(l).Y1 = lbl(xi).Top + lbl(xi).Height / 2
                lin(l).X2 = lbl(yi).Left + lbl(yi).Width / 2
```

الشيفرة البرمجية الكاملة للنموذج frmMain

```
                lin(l).Y2 = lbl(yi).Top + lbl(yi).Height / 2
                lin(l).Visible = True
                lin(l).Tag = xi & ";" & yi
                l = l + 1
            End If
        Next
    Next
Next
End Sub

' عند تغيير النص في مربع النص المخصص لإدخال عناصر المجموعة الأساسية
Private Sub txtElements_Change()

Elements = Split(txtElements.Text, ",")
Num = UBound(Elements) + 1

If Num = 0 Then Exit Sub
' نولد المجموعات الجزئية
GenerateSubsets
' نولد العبارة التي تمثل أسرة المجموعات الجزئية
Dim s As String

s = "{" & Join(Subsets, "}, {") & "}"

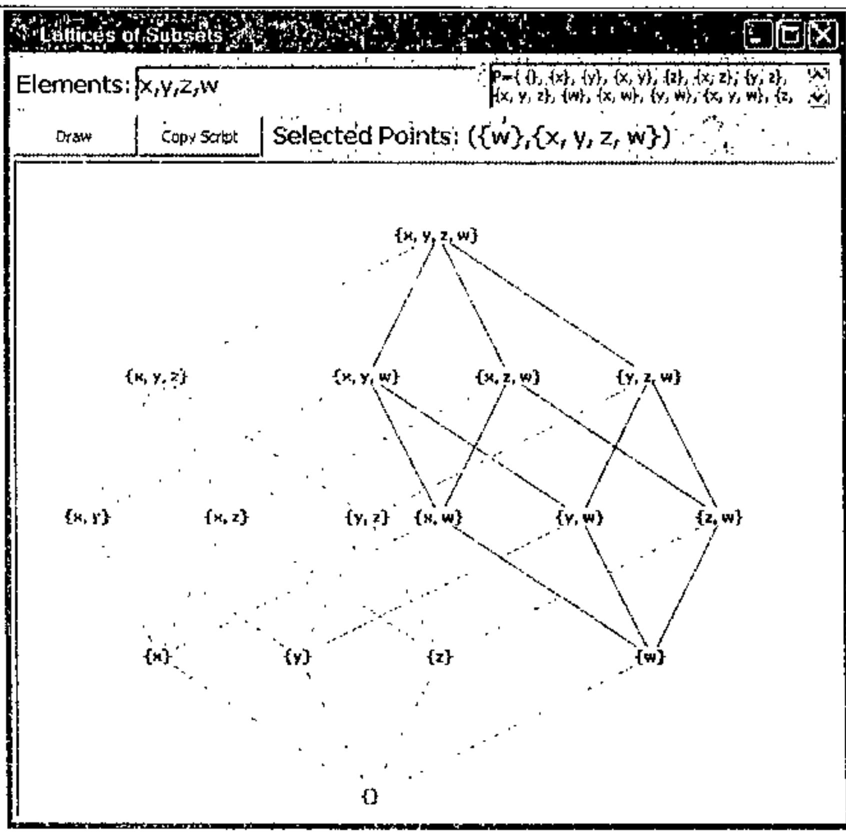
txtPSet.Text = "P={ " & s & " }"
End Sub
```

٤. تنفيذ البرنامج و نتائج الاستخدام Executing and Results

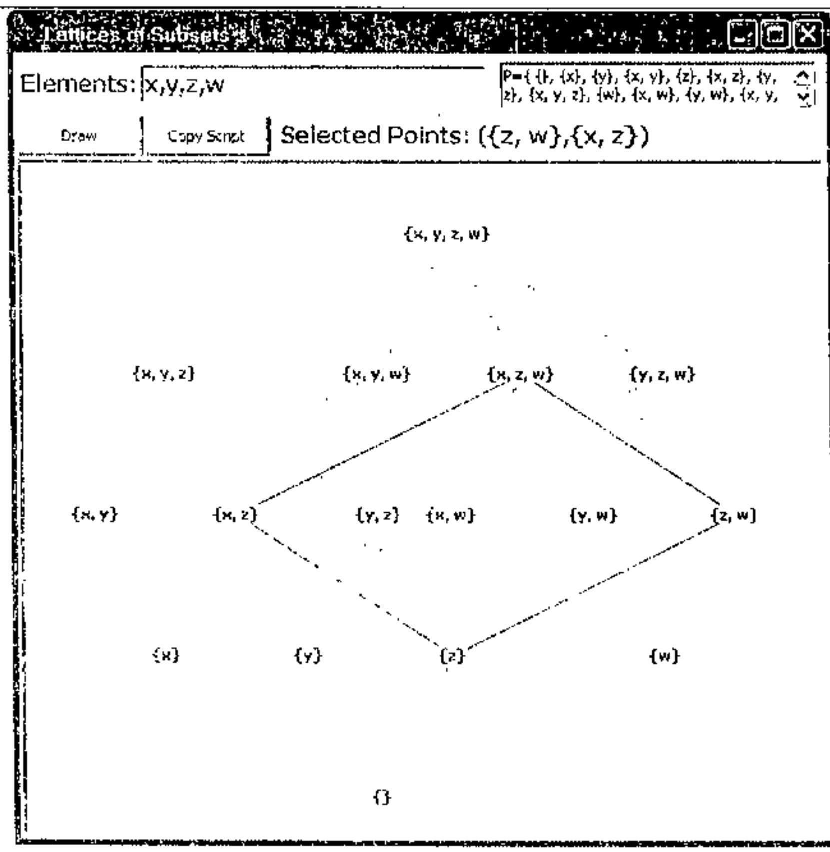
نعرض الآن بعض الأمثلة عن شبكات لمجموعات منتهية قمنا بتطبيقها على البرنامج السابق:

٤-١ مثال. عند إدخال العناصر x, y, z, w في مربع النص الخاص بإدخال عناصر المجموعة يتم تلقائياً توليد أسرة المجموعات الجزئية، و بالنقر على الزر Draw يظهر مخطط هاس للشبكة ({x, y, z, w}, ⊆) ، نقوم بتحسين المخطط عن طريق سحب العناصر. ثم إذا أردنا تحديد المرشحة ({w}] مثلا ننقر بالزر الأيمن على المجموعة {w} ثم ننقر بالزر الأيمن على العنصر الأكبر في الشبكة أي على المجموعة {x, y, z, w} فيتم تلوين المرشحة المطلوبة باللون الأحمر كما يظهر في الشكل ١٢.

ولتحديد الشبكة الجزئية [{x, z}, {z, w}] في الشبكة السابقة بلون مختلف ننقر بالزر الأيمن على المجموعة {x, z} ثم ننقر بالزر الأيمن على المجموعة {z, w} فيتم تلوين الشبكة الجزئية المطلوبة باللون الأحمر كما يظهر في الشكل ١٣.



الشكل ١٢: الشبكة $(P(\{x, y, z, w\}), \subseteq)$ و ضمنها المرشحة $[\ \{w\}\ )$



الشكل ١٣: الشبكة $(P(\{x, y, z, w\}), \subseteq)$ و ضمنها الشبكة الجزئية $[\ \{x, z\}, \{z, w\}\ ]$

§٤ الفصل الرابع

دراسة المجموعات المرتبة
Studying Partly Ordered Sets

١. مقدمة Introduction

نبدأ في هذه الفصل بتحديد مكونات العمل، و معرفة علاقة هذه المكونات بعضها ببعض، تدعى هذه الخطوة بالتحليل العضوي للمسألة. بعد أن نحدد المكونات أو الكيانات و علاقاتها ننتقل إلى دراسة الوظائف الأساسية لكل كيان من هذه الكيانات، وتفاعل هذا الكيان مع البقية. وتدعى هذه الخطوة بالتحليل الوظيفي. ثم ننتقل إلى التصميم الفعلي للبرنامج، فنكتب الشيفرة البرمجية، ونصمم واجهة البرنامج. أخيراً، نختبر البرنامج على حالات محددة و شاملة تمكننا من التأكد من سلامة و صحة العمل الذي يحقق المتطلبات المفروضة في بداية العمل.

١-١ عرض المشكلة.

إن أحد أهدافنا هو نقل الدراسة المجردة لإثبات أن مجموعة مرتبة ما هي شبكة إلى دراسة تطبيقية، أي من الشكل المألوف في الكتب إلى دراسة عن طريق الكمبيوتر. وهنا تظهر عدة مسائل منها كيفية إدخال عناصر المجموعة المرتبة وعلاقة الترتيب عليها كمعطيات إلى البرنامج.

من ناحية ثانية فإن الشبكة تحوي مثاليات و مرشحات و شبكات جزئية علينا استخراجها وتمثيلها بلون مختلف ضمن الشبكة الأصلية. ولا يكتمل البرنامج إلا بقدرته على حفظ المخططات الناتجة في ملف لتعديلها لاحقا.

١-٢ التحليل العضوي.

يتضح لنا بناء على ما سبق أن هناك كيانين أساسيين للمسألة، الأول هو المجموعة والثاني علاقة الترتيب، و ثلاثة كيانات ثانوية هي المثالي و المرشحة والشبكة الجزئية.

ولدى دراسة تفاصيل هذه الكيانات، اكتشفنا عدة قواعد، يجب على البرنامج المطلوب تحقيقها نذكر منها:

١. يفترض البرنامج أن العلاقة المعرفة على المجموعة هي علاقة ترتيب و ينتج عن ذلك القواعد التالية:
   a. يفترض البرنامج أن كل عنصر مرتبط بنفسه.
   b. فقط يتم إدخال علاقة يغطي إلى البرنامج.
   c. يستنتج البرنامج علاقة الترتيب من علاقة يغطي بوساطة خاصة التعدي.
   d. يمنع البرنامج إدخال علاقة يغطي بين عنصرين متقارنين وفق علاقة الترتيب.
   e. يمنع البرنامج إدخال علاقة يغطي بين عنصرين إذا أدى ذلك إلى تعارض مع علاقة يغطي معرفة مسبقاً بين أحد العنصرين و عنصر آخر.
٢. بعد إدخال عناصر الشبكة وعلاقة يغطي عليها يقوم البرنامج باستنتاج علاقة الترتيب عن طريق بناء مجموعتين لكل عنصر، الأولى تحوي العناصر التي يرتبط بها (الأكبر منه)، و الثانية تحوي العناصر التي ترتبط به (الأصغر منه) وفق علاقة الترتيب (انظر ٢§٢-٥).

وضعنا في هذا البحث لغة وصفية خاصة بتعريف المجموعات المرتبة، تتألف هذه اللغة بشكل أساسي من تعليمتين:

١. **تعليمة تعريف العناصر:** كمثال عليها لتعريف المجموعة {x, y, z} تكون التعليمات المكافئة كما يلي:

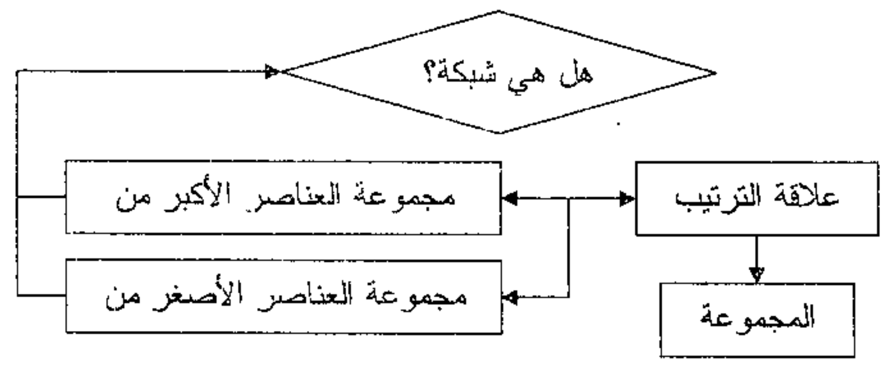
```
v x
v y
v z
```

٢. **تعليمة تعريف علاقة يغطي:** وكمثال عليها بفرض y يغطي x فإن التعليمة المكافئة كما يلي:

```
c 0;1
```

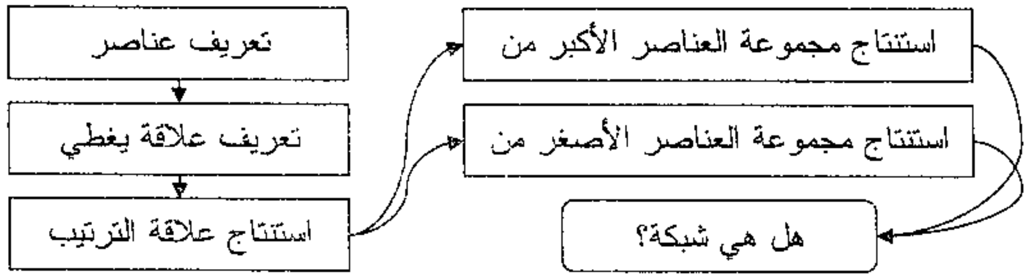
حيث العنصران x و y كما في المثال السابق، نلاحظ أنه تتم إضافة العناصر برموزها بينما تتم إضافة الوصلات بين العناصر عن طريق ذكر ترتيب العنصرين المطلوب إضافة وصلة بينهما مع بدء الترقيم بصفر.

بناء على التحليل السابق يمكن أن نضع مخططا منطقيا لكيانات البرنامج و العلاقات بينها كما يلي:



الشكل ١٤: العلاقات بين كيانات البرنامج

و يمكن تلخيص العمليات الأساسية من خلال المخطط التالي:



الشكل ١٥: العمليات الأساسية

١-٣ التحليل الوظيفي.

وضعنا ست عمليات أساسية يجب أن يقوم بها البرنامج، هذه العمليات هي "إدخال أو تحرير المجموعة المرتبة التي يطلب دراستها، اختبار هل المجموعة المرتبة تشكل شبكة، توليد مخطط هاس تفاعلي للمجموعة المرتبة، تحديد أحد المثاليات أو المرشحات أو المجموعات الجزئية بلون مختلف، حفظ المجموعة المرتبة في ملف، استرجاع المجموعة المرتبة المخزنة في ملف " و تخضع هذه العمليات جميعها إلى القواعد التالية:

١. **إدخال أو تحرير مجموعة مرتبة:**

a. يتم تعريف المجموعة و علاقة الترتيب عليها بوساطة لغة التوصيف السابقة الذكر.

b.   من المهم تذكُر تسلسل تعريف عناصر المجموعة لحاجتنا إلى رقم العنصر عند تعريف العلاقة.

c.   بعد تعريف المجموعة المرتبة يتم رسم مخطط هاس لها مباشرة لأنه الوسيلة المثلى لتمثيلها.

٢. **اختبار "هل هي شبكة؟"**: يتم اختبار هل هي شبكة مباشرة بعد إدخال و إنشاء مخطط المجموعة المرتبة.

٣. **تحديد مرشحة أو مثالي أو مجموعة مرتبة جزئية بلون مختلف:**

a.   تكون المرشحات و المثاليات في الشبكات المنتهية مولدة بعنصر واحد.

b.   يتم تحديد المرشحة بتحديد العنصر المولد و العنصر الأكبر في الشبكة.

c.   يتم تحديد المثالي بتحديد العنصر المولد و العنصر الأصغر.

d.   يتم تحديد المجموعة المرتبة الجزئية بتحديد عنصرين يولدانها و يتم استنتاج علاقة الترتيب عليها تلقائياً.

e.   يتم استخدام الفأرة في عملية التحديد.

٤. **حفظ المجموعة المرتبة في ملف:**

a.   يتم حفظ آخر عمل تلقائياً عند إغلاق البرنامج.

b.   يجب أن يحتوي الملف على تعريف لعناصر المجموعة المرتبة و علاقة يغطى عليها.

c.   بالإضافة إلى ذلك يحتوي الملف على إحداثيات العناصر في مخطط هاس لأن هذه الإحداثيات قابلة للتعديل من قبل مستخدم البرنامج (أي يمكن تحريك عناصر المخطط بسحبها و إفلاتها).

d.   إذا قام المستخدم بتحديد مجموعة مرتبة جزئية بلون مختلف يتم حفظ ذلك ضمن الملف أيضاً.

e.   يتم تحميل الملف تلقائياً عند فتح البرنامج.

٤-١ **بنية البرنامج**. بناء على التحليل السابق للبرنامج نستنتج مكوناته الأساسية وهي:

١. **صنفان (Two Classes)**

a. صنف المجموعة المنتهية: وظيفة هذا الصنف تمثيل مجموعة مؤلفة من عدد محدود من العناصر.

b. صنف المجموعة المرتبة: يعتمد هذا الصنف على الصنف السابق لتمثيل المجموعة المرتبة و إنشاء مخطط هاس الذي يمثلهـا، بالإضـافة إلـى احتوائه على الإجراء الخاص باختبار "هل هي شبكة؟".

٢. وحدة برمجية (One Module)

من ناحية ثانية، لابد من تضمين البرنامج إجراءات تقوم بعمليات حفظ واسترجاع المجموعة المرتبة المدروسة.

٣. نموذجان (Two Forms)

a. نموذج رئيس: يحوي أوامر البرنامج، و يعـرض المخطـط التفـاعلي للمجموعة المدروسة.

b. نموذج إدخال وتحرير: يُمكِّن مستخدم البرنامج من تعريـف المجموعـة المرتبة باستخدام اللغة الوصفية التي أسلفنا ذكرها.

٢. كتابة الأصناف البرمجية Typing Classes

تُعدُّ الأصناف (Classes) حجر الأساس في البرمجة كائنية المنحى (OPP) التي أصبحت في وقتنا الحالي فلسفة البرمجة السائدة التي حلت مكان فلسفة البرمجة الهيكلية (SP).
تنظم فلسفة البرمجة كائنية المنحى الشيفرة البرمجية ضمن أصناف، وهذه الأصناف تجمع بين بنى المعطيات (Data Structures) و سلوك هذه البنى، و تقوم هذه الفلسفة على حسبان كل صنف عبارة عن برنامج مستقل، يمكن استخدامه ضمن أي برنامج آخر بسهولة و مـن دون الدخول في تفاصيل عمله، و يتم ذلك بطريقة سهلة تطابق طريقة الإعلان عـن المتغيـرات، غير أن المتغيرات التي يعلن عنها كصنف تسمى كائنات.
كما أسلفنا في التحليل الوظيفي فنحن بحاجة إلى صنفين: الأول صنف المجموعـة المنتهيـة والثاني يبنى على الأول و هو صنف المجموعة المرتبة.

٢-١. صنف المجموعة المنتهية (Finite Set Class).

إن المجموعة المنتهية تعرف رياضياً بأنها اجتماع لعدد محدود من العناصر. وأول فكرة قـد ترد في الذهن هي تخزين عناصر المجموعة في مصفوفة وحيدة البعد بـدلا مـن اسـتخدام الأصناف البرمجية، لكن بالإضافة إلى هذه المصفوفة نحتاج إلى إجراءات خاصة بإضافة أو

إزالة أو اختبار وجود عنصر ما، لذا كان من الأفضل كتابة كل ذلك كصنف برمجي والتعامل معه ككائن مستقل من دون الدخول في كيفية تمثيل عناصر المجموعة في الذاكرة.

المشكلة الأساسية التي ستعترض تمثيل المجموعات على شكل مصفوفة، هي أن الخوارزميات التي سنعرض لها لاحقاً تعتمد بدرجة كبرى على معاملات اجتماع و تقاطع و فرق المجموعات، و على الرغم من إمكانية كتابة إجراءات تقوم بهذه العمليات على المصفوفة إلا أن ذلك يتطلب الدخول في حلقة تكرارية و إجراء العديد من اختبارات المقارنة، وذلك يزيد كلفة الخوارزمية بشكل كبير.

لذلك رأينا تمثيل عناصر المجموعة على شكل مصفوفة بتات (Bits Array) في الذاكرة، وترقيم عناصر المجموعة الشاملة نسبياً ابتداء من 0، فإذا كانت قيمة أحد البتات (1) فالعنصر المقابل له موجود في المجموعة، أما إذا كانت (0) فالعنصر غير موجود. ويحقق هذا التمثيل هدفنا في كتابة إجراءات خاصّة بإيجاد حاصل اجتماع أو تقاطع أو فرق مجموعتين عن طريق استخدام المعاملات المنطقية الخاصة بالبتات.

٢-١-١ **مثال توضيحي.** لتكن $S=\{0, 1, 2, 3, 4\}$ المجموعة الشاملة نسبياً و المؤلفة من خمسة عناصر، ولا يعنينا شكل أو طبيعة هذه العناصر مادمنا قد رقمناها.
ولتكن المجموعتان $A=\{1, 3, 4\}$ ، $B=\{0, 1\}$ عندئذ يمكن تمثيل المجموعتين وفق ما سبق على النحو:

A					B				
0	1	0	1	1	1	1	0	0	0

و بهذا التمثيل نحصل على اجتماع المجموعتين بوساطة العملية Or، و على التقاطع بوساطة العملية And، و على الفرق بوساطة العمليتين And Not على النحو:

A						B						$A \cup B$				
0	1	0	1	1	OR	1	1	0	0	0	=	1	1	0	1	1

A						B						$A \cap B$				
0	1	0	1	1	AND	1	1	0	0	0	=	0	1	0	0	0

A						B						$A \setminus B$				
0	1	0	1	1	AND NOT	1	1	0	0	0	=	0	0	0	1	1

إذاً تؤول العمليات الأساسية على المجموعات إلى العمليات المنطقية الأساسية على البتات.

وبما أنه لا يمكن حجز متغير في الذاكرة مساحته بت واحد، نقوم بحجز متغير من النوع Long و هو يشغل ٣٢ بتاً من الذاكرة أي يمكننا أن نمثل من خلاله مجموعة مؤلفة على الأكثر من ٣٢ عنصراً، لذلك بدلاً من حجز متغير واحد نعلن عن مصفوفة ديناميكية من النوع Long مما يجعل عدد العناصر الأعظمي غير محدود نظرياً.

فيما يلي نسرد الشيفرة البرمجية الكاملة للصنف clsFiniteSet ثم نفصل بيان كل من الإجراءات الواردة فيها.

الشيفرة البرمجية الكاملة للصنف clsFiniteSet

```
VERSION 1.0 CLASS
BEGIN
  MultiUse = -1  'True
  Persistable = 0  'NotPersistable
  DataBindingBehavior = 0  'vbNone
  DataSourceBehavior  = 0  'vbNone
  MTSTransactionMode  = 0  'NotAnMTSObject
END
Attribute VB_Name = "clsFiniteSet"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = False
Attribute VB_Exposed = False
Option Explicit

Const BNUM = 8 * 4      ' عدد البتات في كل متغير
Const BNMO = BNUM - 1   ' رقم آخر بت إذا بدأنا الترقيم من الصفر

Dim Bits() As Long              ' مصفوفة البتات
Dim PowersOf2(BNMO) As Long   ' مصفوفة تحوي قوى العدد ٢

Dim mCount As Long

' عند تحضير الصنف للعمل
Private Sub Class_Initialize()
InitPowersOf2 ' استدعي الإجراء الذي يولد قوى العدد ٢
End Sub

' الإجراء الخاص بحساب قوى العدد ٢ للاستغناء عن حسابها كلما احتجناها
Private Sub InitPowersOf2()
Dim i As Long
For i = 0 To BNMO - 1
    PowersOf2(i) = 2 ^ i
Next i
PowersOf2(BNMO) = &H80000000
End Sub

' خاصية عدد العناصر في المجموعة الشاملة
Public Property Get MaxCount() As Long
MaxCount = mCount  ' أعد عدد العناصر
End Property

Public Property Let MaxCount(ByVal NewValue As Long)
```

```
If NewValue <= 0 Then
    ' مطلوب مجموعة فارغة
    Erase Bits
    mCount = 0
Else
    ' أعد تحجيم مصفوفة البتات كي تتسع للعدد المطلوب من العناصر
    Dim tb As Long
    tb = (NewValue - 1) \ BNUM
    mCount = tb * BNUM + BNUM
    ReDim Preserve Bits(tb)
End If
End Property

' الطريقة الخاصة بإضافة عنصر إلى المجموعة
Public Sub AddElement(ByVal X As Long)
If X < 0 Then Err.Raise 5, , "Element cannot be negative."
If X >= mCount Then Me.MaxCount = X + 1
' أضبط البت الذي يمثل العنصر بواحد
SetBit X, True
End Sub

' الطريقة الخاصة بإزالة عنصر
Public Sub RemElement(ByVal X As Long)
If X < 0 Then Err.Raise 5, , "Element cannot be negative."
' أضبط البت الذي يمثل العنصر بصفر
If X < mCount Then SetBit X, False
End Sub

' هل العنصر ينتمي إلى المجموعة
Public Function IsElement(ByVal X As Long) As Boolean
If X < 0 Then Err.Raise 5, , "Element cannot be negative."
If X >= mCount Then
    IsElement = False
Else
    ' أعد قيمة البت الذي يمثل العنصر
    IsElement = TestBit(X)
End If
End Function

' الإجراء الخاص بضبط قيمة أحد البتات ضمن مصفوفة البتات
Private Sub SetBit(ByVal Bit As Long, ByVal Value As Boolean)
If Bit < 0 Then Call Err.Raise(5, , "Bit cannot be negative.")

Dim Index As Long
' إلى أي عنصر في المصفوفة ينتمي البت المطلوب
Index = Bit \ BNUM
If Value Then
    ' نضع في البت واحداً بمساعدة مصفوفة قوى العدد ٢
    Bits(Index) = Bits(Index) Or PowersOf2(Bit And BNMO)
Else
    ' نضع في البت صفراً
    Bits(Index) = Bits(Index) And (Not PowersOf2(Bit And BNMO))
End If
End Sub

' ما هي قيمة أحد البتات
Private Function TestBit(ByVal Bit As Long) As Boolean
Dim Index As Long
'حدد العنصر الذي ينتمي إليه البت المطلوب
```

```
Index = Bit \ BNUM
'نختبر بمساعدة مصفوفة قوى ٢'
TestBit = Bits(Index) And (PowersOf2(Bit And BNMO))
End Function

' خاصية تعيد مصفوفة البتات و تفيد في حساب التقاطع و الاجتماع و الفرق
Public Property Get GetBits() As Long()
GetBits = Bits
End Property

' الإجراء الخاص بعملية إسناد مجموعة إلى المجموعة الحالية
Public Sub LetSet(S As clsFiniteSet)
' نجعل عدد العناصر و مصفوفة البتات مساوية لمقابلاتها
Me.MaxCount = S.MaxCount
Bits = S.GetBits
End Sub

' الإجراء الخاص بعمليه الاجتماع
Public Sub JoinSet(S As clsFiniteSet)
Dim SBits() As Long, i As Long

If S.Count = 0 Then Exit Sub
' نعدل عدد العناصر كي يتسع لعناصر المجموعتين
If S.MaxCount > Me.MaxCount Then Me.MaxCount = S.MaxCount
' نطبق العملية "أو" على بتات المجموعتين
SBits = S.GetBits
For i = 0 To UBound(SBits)
    Bits(i) = Bits(i) Or SBits(i)
Next
End Sub

' الإجراء الخاص بعملية التقاطع
Public Sub MeetSet(S As clsFiniteSet)
Dim SBits() As Long, i As Long

If S.Count = 0 Then
    Me.MaxCount = 0
    Exit Sub
End If
If S.MaxCount < Me.MaxCount Then Me.MaxCount = S.MaxCount
' نطبق العملية "و" على بتات المجموعتين
SBits = S.GetBits
For i = 0 To UBound(Bits)
    Bits(i) = Bits(i) And SBits(i)
Next
End Sub

' الإجراء الخاص بعملية الفرق
Public Sub SubtractSet(S As clsFiniteSet)
Dim SBits() As Long, i As Long, tb As Long

If S.MaxCount = 0 Then Exit Sub

tb = (IIf(S.MaxCount > mCount, mCount, S.MaxCount) - 1) \ BNUM
' نطبق العمليتين "و ليس" على بتات المجموعتين
SBits = S.GetBits
For i = 0 To tb
    Bits(i) = Bits(i) And Not SBits(i)
Next
```

الشيفرة البرمجية الكاملة للصنف clsFiniteSet

```
End Sub

' خاصية تعيد أحد العناصر التي تنتمي إلى المجموعة
Public Property Get Element(ByVal Index As Long) As Long
Attribute Element.VB_UserMemId = 0  ' هي الخاصية الافتراضية
Dim i As Long, j As Long

Element = -1
i = 0
j = -1
' نختبر بتات المجموعة واحداً واحداً حتى نحصل على رقم العنصر
Do While i < mCount
    If TestBit(i) Then j = j + 1
    If j = Index Then
        Element = i
        Exit Property
    End If
    i = i + 1
Loop
End Property

' خاصية تعيد كل العناصر التي تنتمي إلى المجموعة وهي مفيدة في الاختبار
Public Property Get GetElements() As Long()
Dim i As Long, j As Long, e() As Long

i = 0
j = -1
Do While i < mCount
    If TestBit(i) Then
        j = j + 1
        ReDim Preserve e(j)
        e(j) = i
    End If
    i = i + 1
Loop
GetElements = e
End Property

' خاصية تحسب عدد العناصر التي تنتمي إلى المجموعة
Public Property Get Count() As Long
Dim i As Long
' نختبر من أول بت إلى آخر بت و نحسب عدد البتات التي قيمتها ١
Count = 0
For i = 0 To mCount - 1
    If TestBit(i) Then Count = Count + 1
Next
End Property
```

٦٨٠٦٣٤

يبدأ الصنف بالإعلان عن ثابتين الأول هو BNUM و يمثل عدد البتات في كل متغير من نوع Long والثاني BNMO و يمثل رقم آخر بت في المتغير عندما نبدأ الترقيم من الصفر، أي BNUM=32 و BNMO=31، ثم يأتي الإعلان عن المصفوفة الديناميكية Bits() من النوع Long التي سنستخدمها لتمثيل المجموعة المنتهية. ثم نعلن عن المصفوفة PowersOf2(31) من النوع Long و التي نخزن فيها قوى العدد 2 بدءاً من 2^0 و حتى

2^{31} لحاجتنا إليها في الإجراءات اللاحقة. بعد ذلك نعلن عن mCount من النوع Long وسنستعمله لتخزين عدد العناصر الأعظمي.

يتم استدعاء الإجراء Class_Initialize تلقائياً عند وقوع الحدث Initialize الخاص بالصنف، و يقع هذا الحدث عند الإعلان عن متغير من نوع الصنف مما يسمح بإجراء أية عمليات تحضيرية نحتاج إليها. و كما هو واضح فقد استعملنا هذا الحدث لاستدعاء الإجراء InitPowersOf2 الذي يقوم بحساب قوى العدد 2 ويخزنها في المصفوفة PowersOf2(31) التي أسلفنا ذكرها.

إن الخاصية MaxCount تضبط أو تعيد العدد الأعظمي للعناصر التي يمكن للمجموعة المنتهية أن تحويها، و بما أن كل عنصر يمثل على شكل بت، ولدينا كل 32 بتاً في متغير واحد من النوع Long، لذا إما أن تأخذ الخاصية MaxCount القيمة 0 و يتم إلغاء مساحة الذاكرة المخصصة للمصفوفة الديناميكية عن طريق التعليمة Erase Bits و إما أن يتم ضبط الخاصية MaxCount بأصغر مضاعف للعدد 32 يكبر العدد الإجمالي للعناصر، كما يتم تغير عدد العناصر في المصفوفة ()Bits ليصبح حاصل القسمة الصحيحة للعدد المطلوب من العناصر على 32، وذلك من خلال الأسطر الثلاثة التالية:

```
tb = (NewValue - 1) \ BNUM
mCount = tb * BNUM + BNUM
ReDim Preserve Bits(tb)
```

إن الإجراء AddElement هو طريقة (Method) خاصة بالصنف وظيفتها وضع القيمة 1 البت الذي يمثل العنصر X، على عكس الإجراء RemElement الذي يضع 0 في البت الذي يمثل العنصر X، و لا يؤثر استدعاء AddElement أكثر من مرة لإضافة العنصر نفسه، لأنه سيتم ضبط البت نفسه بالقيمة 1، مما يمنع تكرار العنصر نفسه ضمن المجموعة و يلغي الحاجة إلى التحقق من كون العنصر موجوداً مسبقاً قبل إضافته، مما يوفر زمن التنفيذ.

إنّ كلا الإجراءين يستدعي الإجراء SetBit الذي يقبل وسيطين اثنين، الأول هو رقم البت المطلوب ضبط قيمته، و الثاني إما True إذا كان المطلوب ضبط البت بالقيمة 1، و إما False إذا كانت القيمة المطلوبة هي 0.

يقوم الإجراء SetBit أولاً بتحديد موضع البت ضمن المصفوفة ()Bits من خلال القسمة الصحيحة على 32، ثم يبرز دور المصفوفة (PowersOf2(31. للتوضيح انظر الجدول التالي الذي يسرد قوى 2 مكتوبة في النظام الثنائي:

$2^0 = 1 = (1)_2$
$2^1 = 2 = (01)_2$
$2^2 = 4 = (001)_2$
$2^3 = 8 = (0001)_2$
$2^4 = 16 = (00001)_2$
......

أي أن العدد 2^x يكتب بالنظام الثنائي على شكل عدد جميع خاناته أصفار إلا الخانة التي رقمها x إذا بدأنا بترقيم الخانات من الصفر. فإذا أردنا أن نضبط البت ذا الرقم x بالقيمة 1 يكفي أن نطبق العملية Or على مصفوفة البتات و العدد 2^x وهذا ما يقوم به السطر:

Bits(Index) = Bits(Index) Or PowersOf2(Bit And BNMO)

أما إذا أردنا ضبط البت بالقيمة 0 فيجب تطبيق العمليتين Not And بدلاً من Or و هذا ما يقوم به السطر:

Bits(Index) = Bits(Index) And (Not PowersOf2(Bit And BNMO))

الدالة TestBit تختبر القيمة الموجودة في البت ذي الرقم X بشكل مشابه، إذ يكفي تطبيق العملية And كما في السطر:

TestBit = Bits(Index) And (PowersOf2(Bit And BNMO))

الدالة IsElement تستدعي الدالة TestBit لمعرفة إن كانت المجموعة محتوية على العنصر ذا الرقم X أي إذا كان البت الموافق يحوي القيمة 1.

الآن نصل إلى الإجراءات الأساسية التي تشكل الطرق (Methods) الخاصة بإسناد و اجتماع و تقاطع و فرق المجموعات و فيما يلي أمثلة على عمل و طريقة استخدام كل منها:

A.LetSet B	إسناد عناصر المجموعة B إلى المجموعة A أي يصبح $A = B$
A.JoinSet B	إسناد اجتماع A و B إلى A أي يصبح $A = A \cup B$
A.MeetSet B	إسناد تقاطع A و B إلى A أي يصبح $A = A \cap B$
A.SubtractSet B	إسناد A فرق B إلى A أي يصبح $A = A \setminus B$

إن كلاً من الإجراءات السابقة يستدعى الدالة ()GetBits التي تعود بمصفوفة البتات الخاصة بالمجموعة الممررة كوسيط ثم تطبق العمليات المنطقية على مصفوفة المجموعة الممررة ومصفوفة المجموعة الأساسية.

يسند الإجراء LetSet عدد العناصر الأعظمي المجموعة الممررة إلى المجموعة الأساسية ثم يضبط مصفوفة البتات في المجموعة الأساسية لكي تساوي مقابلتها في المجموعة الممررة، و بالتالي تصبح المجموعتان متساويتين.

أما الإجراءات JoinSet و MeetSet و SubtractSet فتطبق العمليات المنطقية التي أسلفنا شرحها على مصفوفة بتات المجموعتين الأساسية و الممررة لإيجاد حاصل اجتماع وتقاطع وفرق المجموعتين من خلال الأسطر التالية على الترتيب:

```
Bits(i) = Bits(i) Or SBits(i)
Bits(i) = Bits(i) And SBits(i)
Bits(i) = Bits(i) And Not SBits(i)
```

مع ملاحظة أنه في حالة الاجتماع يتم تكبير عدد العناصر الأعظمي في المجموعة الأساسية إذا كان مقابله في المجموعة الممررة أكبر ليصبح مساويا له، أما في حالة التقاطع فيتم تصغيره إذا كان مقابله أصغر، و في حالة الفرق لا يتم التكبير أو التصغير، فقط يتم أخذ أصغر العددين كحد أعلى لحلقة For التي تقوم بالعمليات المنطقية السابقة.

إن الخاصية Element تمثل الخاصية الافتراضية للصنف clsFiniteSet و ترجع أول عنصر رقمه Index ينتمي إلى المجموعة مع مراعاة ترقيم العناصر من الصفر.
إن كون الخاصية افتراضية يعني على سبيل المثال أن العبارتين:

```
x = A.Element(3)
x = A(3)
```

متكافئتان أي يمكن الاستغناء عن ذكر اسم الخاصية بعد اسم الكائن، و السطر المسؤول عن جعل الخاصية افتراضية في الصنف هو

```
Attribute Element.VB_UserMemId = 0
```

تعمل الخاصية Element على اختبار البتات من أول بت و حتى آخر بت عن طريق استدعاء الدالة TestBit مع حساب عدد البتات التي تحوي القيمة 1، و عند الوصول إلى عدد بتات يساوي Index فإنها تعيد رقم البت، و إلا فإنها تعيد القيمة 1-.

إن عمل الخاصية GetElements مشابه لسابقتها غير أنها تعيد جميع العناصر على شكل مصفوفة بدلا من إعادة عنصر واحد، و على الرغم من أننا لم نستعملها في البرنامج لاحقاً إلا أننا استفدنا منها في مراقبة (Watch) سلوك المجموعة واختبارها في طور التصميم لأنه

يمكن إضافة هذه الخاصية إلى قائمة المتغيرات المراقبة في بيئة التطوير مما يتيح معرفة العناصر التي تحتويها المجموعة لحظة بلحظة.

أخيراً فإن الخاصية Count تعيد عدد العناصر التي تحويها المجموعة أي تعيد عدد البتات التي تحوي القيمة 1 عن طريق استدعاء الإجراء TestBit ضمن حلقة تكرارية من أول بت إلى آخر بت.

خلاصة القول إن صنف المجموعة المنتهية يحل مسألة تمثيل المجموعات المجردة المنتهية، مع الانتباه إلى أن العناصر تؤخذ بأرقامها و ليس بذاتها، حيث إن أخذ العناصر بصيغتها المجردة كأرقام فقط لن يؤثر في حل مسائل الشبكات لأننا إذا غيرنا رمز أيّ عنصر أو اسمه من المجموعة مع المحافظة على علاقاته مع بقية العناصر فلن يؤثر ذلك على طبيعة الشبكة وخواصها. لكننا حللنا مسألة ظهور العناصر برموزها بدلاً من أرقامها في مخطط هاس الذي يولده البرنامج كما سنرى لاحقاً في تصميم صنف المجموعة المرتبة.

٢-٢ صنف المجموعة المرتبة (Ordered Set Class).

يقوم هذا الصنف بمعظم الأمور التي ناقشناها في تحليل البرنامج و تقع عليه مسؤولية التعامل مع كل ما يختص بالمجموعات المرتبة.

ذكرنا في ٤§١-٢ أننا بحاجة إلى المجموعتين [x) و (x] لذا يتوجب علينا توليد هاتين المجموعتين لكل عنصر فور إضافته إلى المجموعة، و من ثم تعديل هذه المجموعات عند إضافة علاقة يغطي (وصلة) بين عنصرين.

ولذلك عند تعريف علاقة y يغطي x نقوم أولا بتنفيذ الخوارزمية التالية:

٢-٢-١ خوارزمية.

١. من أجل كل s من [y)

أ. $(s] \leftarrow (s] \cup (x]$

٢. من أجل كل r من (x]

أ. $[r) \leftarrow [r) \cup [y)$

و هذه العملية تضمن أن المجموعتين السابقتين في جاهزية تامة دائماً ولا حاجة لإعادة حسابها كلما اقتضى الأمر، لأنها تدخل في حل جميع المسائل التي ستواجهنا.

أولى هذه المسائل تكمن في تحديد مجموعتي الحدود العليا و الحدود الدنيا لأي عنصرين x و y من المجموعة المرتبة، ويمكن ذلك بسهولة وفق ما يلي:

مجموعة الحدود العليا للعنصرين x و y: $U(x,y) = [\, x\,) \cap [\, y\,)$

مجموعة الحدود الدنيا للعنصرين x و y: $L(x,y) = (\, x\,] \cap (\, y\,]$

نقدم الآن مبرهنتين أساسيتين تمكنانا من حساب الحدين الأعلى الأصغري و الأدنى الأعظمي لأي عنصرين في المجموعة المرتبة جزئياً.

٢-٢-٢ مبرهنة. لتكن المجموعة $S_{x,y} = U(x,y) \cap (\bigcap_{z\in U(x,y)} (\, z\,])$ عندئذ إما $S_{x,y} = \varnothing$ وإما مجموعة تحوي عنصراً وحيداً هو $Sup(x,y)$.

البرهان. لنبرهن أنه إذا كان $s \in S_{x,y}$ فإن $s = Sup(x,y)$، من تعريف المجموعة $S_{x,y}$ نجد أن $s \in U(x,y)$ و $s \in \bigcap_{z\in U(x,y)} (\, z\,]$، إذن s حد أعلى للعنصرين x و y لأنه ينتمي إلى مجموعة الحدود العليا لهما، وبفرض r حداً أعلى آخر للعنصرين x و y عندئذ $s \in \bigcap_{z\in U(x,y)} (\, z\,]$ وبالتالي $s \in (\, r\,]$ وبحسب تعريف $(\, x\,]$ نجد أن $s \le r$ و منه $s = Sup(x,y)$. أصبح لدينا كل عنصر من $S_{x,y}$ هو $Sup(x,y)$، وبما أن $Sup(x,y)$ إن وجد فهو وحيد، فالمجموعة $S_{x,y}$ تحقق الادعاء.

بالأسلوب نفسه يمكن البرهان على صحة ما يلي:

٢-٢-٣ مبرهنة. لتكن المجموعة $N_{x,y} = L(x,y) \cap (\bigcap_{z\in L(x,y)} [\, z\,))$ عندئذ إما $N_{x,y} = \varnothing$ وإما مجموعة تحوي عنصراً وحيداً هو $Inf(x,y)$.

الآن يمكننا كتابة خوارزمية تجيب على السؤال التالي "هل المجموعة المرتبة المعطاة تشكل شبكة أم لا؟".

٢-٢-٤ خوارزمية. إن التحقق من أن مجموعة مرتبة هي شبكة يتضمن اختبار وجود الحدين الأعلى الأصغري و الأدنى الأعظمي لكل زوج من عناصر الشبكة، لذلك نكتب الخوارزمية التالية:

```
١. من أجل كل x من المجموعة المرتبة
    أ. من أجل كل y من المجموعة المرتبة
        ١. إذا كان Sup(x,y) أو Inf(x,y) غير موجودين نفذ
            أ. اطبع ليست شبكة بسبب x و y
            ب. اخرج
```

٢. اطبع هي شبكة

إن الخوارزمية السابقة تتصف بعيب أساسي سببه هو أن $Sup(x,y) = Sup(y,x)$ وبالتالي سيتم إجراء اختبار وجود الحدين الأعلى الأصغري و الأدنى الأعظمي لكل زوج من العناصر مرتين، بالإضافة إلى عدم حاجة اختبار العناصر المتقارنة وفق علاقة الترتيب لأنه إذا كان $x \leq y$ فإن $Sup(x,y) = y$ و $Inf(x,y) = x$ (انظر ١§٣-٢) ولحل هذه المشكلة عدلنا الخوارزمية كما يلي:

١. من أجل كل x من المجموعة المرتبة S

أ. $T_x \leftarrow T_x \cup (x] \cup [x)$

ب. $R \leftarrow S \setminus T_x$

ج. من أجل كل y من R

١. إذا كان $Sup(x,y)$ أو $Inf(x,y)$ غير موجودين نفّذ

أ. اطبع ليست شبكة بسبب x و y

ب. اخرج

٢. $T_y \leftarrow T_y \cup \{x\}$

٢. اطبع هي شبكة

هذه الخوارزمية تتلافى ما سبق من عيوب، فهي تختبر وجود الحدين فقط للعناصر غير المتقارنة التي لم يتم اختبارها سابقاً وذلك عن طريق المجموعة T_x التي نخزن فيها جميع العناصر التي لا حاجة لمقارنتها بالعنصر x وهي العناصر الموجودة في [x) و (x] والعناصر التي تمت مقارنتها سابقاً.

نصل الآن إلى مسألة إنشاء مخطط هاس للمجموعة المرتبة، إن المخطط المطلوب مرتبط ارتباطاً جذرياً بعلاقة يُغطّي، في هذا المخطط سنستبدل عبارة y يغطي x بالعبارة y متصل بـ x، وسنرمز بالرمز IC_x لمجموعة العناصر التي يتصل بها x (يغطيها x)، و بالرمز FC_x لمجموعة العناصر المتصلة بالعنصر x (تغطي x).

بالنظر إلى أي مخطط نجد أن عناصره يمكن تجميعها ضمن أسطر، بحيث إن عناصر كل سطر تكون أكبر من عناصر الأسطر التي تحتها و أصغر من عناصر السطر الذي فوقها وفق علاقة الترتيب المعرفة.

وضعنا الخوارزمية التالية التي تستنتج مجموعات السطور row_i حيث i رقم السطر بحيث إن مجموعة السطر الذي يقع أسفل السطور هي row_0، و مجموعة السطر الذي فوقه مباشرة هي row_1 ...

٢-٢-٥ خوارزمية.

١. لتكن S المجموعة المرتبة

٢. $row_0 \leftarrow S$

٣. $i \leftarrow 0$

٤. كرر

أ. من أجل كل x من row_i

١. من أجل كل y من FC_x

أ. $row_{i+1} \leftarrow row_{i+1} \cup \{y\}$

ب. من أجل كل z من row_{i+1}

١. $row_i \leftarrow row_i \setminus \{z\}$

ج. $i \leftarrow i + 1$

٥. حتى يصبح $|row_i| = 0$

نفترض في بداية الخوارزمية أن جميع العناصر تقع في سطر واحد، ثم نرفع جميع العناصر المتصلة بعناصر أخرى إلى السطر التالي، و بعدها ننتقل إلى السطر التالي و نكرر العملية حتى لا تبقى عناصر في السطر الذي وصلنا إليه.

نقدم فيما يلي الشيفرة البرمجية الكاملة للصنف clsOrderedSet ثم نفصل بيان كل من الإجراءات الواردة فيها.

الشيفرة البرمجية الكاملة للصنف clsOrderedSet

```
VERSION 1.0 CLASS
BEGIN
  MultiUse = -1  'True
  Persistable = 0  'NotPersistable
  DataBindingBehavior = 0  'vbNone
  DataSourceBehavior  = 0  'vbNone
  MTSTransactionMode  = 0  'NotAnMTSObject
END
Attribute VB_Name = "clsOrderedSet"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = False
Attribute VB_Exposed = False
Option Explicit

Dim sLastError As String

Dim VSet As New clsFiniteSet  ' المجموعة المنتهية
Dim VI() As New clsFiniteSet  ' (x]
Dim VF() As New clsFiniteSet  ' [x)
Dim VIC() As New clsFiniteSet ' ICx
Dim VFC() As New clsFiniteSet ' FCx
```

الشيفرة البرمجية الكاملة للصنف clsOrderedSet

```
Dim ElementsAlias() As String ' تسميات عناصر المجموعة

' الطريقة الخاصة بإضافة عنصر إلى المجموعة
Public Sub AddVertex(Optional Alias)
Dim X As Long
' يساوي عدد عناصر المجموعة عند بدء الترقيم من الصفر
X = VSet.Count
VSet.AddElement X   ' نضيف العنصر إلى المجموعة
' نوسع المصفوفات
ReDim Preserve VI(X)
ReDim Preserve VF(X)
ReDim Preserve VIC(X)
ReDim Preserve VFC(X)
ReDim Preserve ElementsAlias(X)
' كل عنصر يرتبط بنفسه لذلك نضيفه مباشرة إلى مجموعتي العناصر الأكبر
' منه و الأصغر منه
VI(X).AddElement X
VF(X).AddElement X
ElementsAlias(X) = IIf(IsMissing(Alias), X, Alias) ' خزن تسمية العنصر
End Sub

' اختبار ارتباط عنصرين
Public Function CheckRelation(X As Long, Y As Long) As Boolean
CheckRelation = VI(Y).IsElement(X)
End Function

' الإجراء الخاص بإضافة وصلة بين عنصرين
Public Function AddConnection(X As Long, Y As Long) As Boolean
Dim i As Integer

AddConnection = False
' هل العنصران ضمن عناصر المجموعة
i = VSet.Count - 1
If X < 0 Or X > i Or Y < 0 Or Y > i Then Exit Function
' هل العنصران مرتبطان
If CheckRelation(X, Y) Or CheckRelation(Y, X) Then
    sLastError = ElementsAlias(X) & " and " & _
                 ElementsAlias(Y) & " already in relation"
    Exit Function
End If
' هل ستؤدي إضافة هذه الوصلة إلى تعارض مع وصلة أخرى
For i = 0 To VFC(X).Count - 1
    If CheckRelation(Y, VFC(X)(i)) Then
        ' سجل الخطأ
        sLastError = "connot connect " & ElementsAlias(X) & _
                     " to " & ElementsAlias(Y)
        Exit Function
    End If
Next
For i = 0 To VIC(Y).Count - 1
    If CheckRelation(VIC(Y)(i), X) Then
        ' سجل الخطأ
        sLastError = "connot connect " & ElementsAlias(X) & _
                     " to " & ElementsAlias(Y)
        Exit Function
    End If
Next
' أضف الوصلة بين العنصرين
VFC(X).AddElement Y
```

```
VIC(Y).AddElement X
' نعيد حساب مجموعتي العناصر (الأكبر من) و (الأصغر) من لكل العناصر المتأثرة '
For i = 0 To VF(Y).Count - 1
    VI(VF(Y)(i)).JoinSet VI(X)
Next
For i = 0 To VI(X).Count - 1
    VF(VI(X)(i)).JoinSet VF(Y)
Next

AddConnection = True
End Function

' إجراء حساب مجموعة الحدود العليا لعنصرين '
Public Function Max(X As Long, Y As Long) As clsFiniteSet
Dim res As New clsFiniteSet

res.LetSet VF(X)  ' res=VF(x)∩VF(y)
res.MeetSet VF(Y)

Set Max = res
End Function

' إجراء حساب مجموعة الحدود الدنيا لعنصرين '
Public Function Min(X As Long, Y As Long) As clsFiniteSet
Dim res As New clsFiniteSet

res.LetSet VI(X)  ' res=VI(x)∩VI(y)
res.MeetSet VI(Y)

Set Min = res
End Function

' حساب الحد الأعلى الأصغري لعنصرين '
Public Function Sup(X As Long, Y As Long) As Long
Dim Mxy As clsFiniteSet, res As New clsFiniteSet
Dim n As Long, i As Long

Sup = -1

Set Mxy = Max(X, Y)
n = Mxy.Count
If n = 1 Then
    Sup = Mxy(0)
    Exit Function
End If

res.LetSet Mxy
For i = 0 To n - 1
    res.MeetSet VI(Mxy(i))
Next
' res=Sxy
If res.Count = 1 Then Sup = res(0)

End Function

' حساب الحد الأدنى الأعظمي لعنصرين '
Public Function Inf(X As Long, Y As Long) As Long
Dim Mxy As clsFiniteSet, res As New clsFiniteSet
Dim n As Long, i As Long
```

```
Inf = -1

Set Mxy = Min(X, Y)
n = Mxy.Count
If n = 1 Then
    Inf = Mxy(0)
    Exit Function
End If

res.LetSet Mxy
For i = 0 To n - 1
    res.MeetSet VF(Mxy(i))
Next
' res=Nxy
If res.Count = 1 Then Inf = res(0)

End Function

' اختبار هل هي شبكة.
Public Function IsLattice() As Boolean
Dim res As New clsFiniteSet
Dim tested() As New clsFiniteSet
Dim X As Long, Y As Long, i As Long, n As Long

n = VSet.Count
IsLattice = True

If n = 0 Then Exit Function

ReDim tested(n - 1)

For X = 0 To n - 1
    res.LetSet VSet
    res.SubtractSet VI(X)
    res.SubtractSet VF(X)
    res.SubtractSet tested(X)
    For i = 0 To res.Count - 1
        Y = res(i)
        If Sup(X, Y) = -1 Or Inf(X, Y) = -1 Then
            ' سجل سبب ليست شبكة
            sLastError = "Not Lattice because " & ElementsAlias(X) _
                        & " and " & ElementsAlias(Y)
            IsLattice = False
            Exit Function
        End If
        tested(Y).AddElement X
    Next
Next
End Function

' الإجراء الخاص بتوليد مخطط هاس للمجموعة المرتبة
Public Sub GeneratLattice(lin As Object, lbl As Object)
Dim rows() As New clsFiniteSet
Dim X As Long, Y As Long
Dim i As Long, j As Long, k As Long, m As Long, n As Long, l As Long

' نطبق الخوارزمية لمعرفة عناصر أسطر المخطط
ReDim rows(0)
```

```
rows(0).LetSet VSet

m = 0
i = 0
Do
    ReDim Preserve rows(i + 1)
    For j = 0 To rows(i).Count - 1
        X = rows(i)(j)
        For k = 0 To VFC(X).Count - 1
            Y = VFC(X)(k)
            rows(i + 1).AddElement Y
        Next
    Next
    For j = 0 To rows(i + 1).Count - 1
        X = rows(i + 1)(j)
        rows(i).RemElement X
    Next
    ' هل هذا السطر هو أكثر سطر يحوي عناصر
    If rows(i).Count > rows(m).Count Then m = i
    i = i + 1
Loop While rows(i).Count

ReDim Preserve rows(i - 1)

' نحمل اللافتات و الخطوط المطلوبة و نزيل الزائدة
n = VSet.Count
l = 0
For i = 0 To n - 1
    l = l + VFC(i).Count
Next

For i = n To lbl.Count - 1
    Unload lbl(i)
Next
For i = l To lin.Count - 1
    Unload lin(i)
Next
For i = lbl.Count To n - 1
    Load lbl(i)
Next
For i = lin.Count To l - 1
    Load lin(i)
Next

Const SPA = 100
Dim xn As Single, yn As Single, X1 As Single, Y1 As Single

xn = 100
yn = (UBound(rows) - m) * SPA

' نبدأ من أطول سطر إلى آخر سطر
For i = m To UBound(rows)
    Y1 = yn - (i - m) * SPA
    For j = 0 To rows(i).Count - 1
        ' نحرك اللافتات إلى مواضعها و نضبط عناوينها
        X = rows(i)(j)
        lbl(X).Caption = ElementsAlias(X)
        X1 = xn + (rows(m).Count - rows(i).Count) * SPA / 2 + _
                j * SPA - lbl(X).Width / 2
```

```
        lbl(X).Move X1, Y1
        lbl(X).Visible = True
        lbl(X).ZOrder
    Next
Next
' الآن من السطر الذي تحت أطول سطر إلى أول سطر
For i = m - 1 To 0 Step -1
    Y1 = yn + (m - i) * SPA
    For j = 0 To rows(i).Count - 1
        ' تحرك اللافتات إلى مواضعها و نضبط عناوينها
        X = rows(i)(j)
        lbl(X).Caption = ElementsAlias(X)
        X1 = xn + (rows(m).Count - rows(i).Count) * SPA / 2 + _
              j * SPA - lbl(X).Width / 2
        lbl(X).Move X1, Y1
        lbl(X).Visible = True
        lbl(X).ZOrder
    Next
Next
' تحرك الخطوط إلى مكانها مستخدمين علاقة يغطي
m = 0
For X = 0 To n - 1
    For k = 0 To VFC(X).Count - 1
        Y = VFC(X)(k)
        lin(m).X1 = lbl(X).Left + lbl(X).Width / 2
        lin(m).X2 = lbl(Y).Left + lbl(Y).Width / 2
        lin(m).Y1 = lbl(X).Top + lbl(X).Height / 2
        lin(m).Y2 = lbl(Y).Top + lbl(Y).Height / 2
        lin(m).Tag = X & ";" & Y
        lin(m).Visible = True
        m = m + 1
    Next
Next
End Sub

' خاصية عدد العناصر
Public Property Get Count() As Long
Count = VSet.Count
End Property

' خاصة الخطأ الأخير التي تعيد آخر خطأ حصل
Public Property Get LastError() As String
LastError = sLastError
End Property

' خاصة تسمية العنصر
Public Property Get Alias(X As Long) As String
Dim n As Long

n = VSet.Count - 1
If X < 0 Or X > n Or Y < 0 Or Y > n Then Exit Function
Alias = ElementsAlias(X)
End Property

' الطريقة الخاصة بمسح عناصر المجموعة
Public Sub ClearSet()
Erase VI
Erase VF
Erase VIC
```

الشيفرة البرمجية الكاملة للصنف clsOrderedSet

```
Erase VFC
Erase ElementsAlias
VSet.MaxCount = 0
End Sub
```

كما تلاحظ فكل من الإجراءات السابقة هو انعكاس للخوارزميات السابقة بعبارات برمجية، الإجراء الذي يحتاج إلى تفصيل هو الإجراء GeneratLattice الذي يقوم بإنشاء مخطط هاس للمجموعة المرتبة.

نبدأ بتطبيق الخوارزمية التي ذكرناها لتحديد عناصر كل سطر و لمعرفة عدد الأسطر المطلوب في المخطط، وخلال هذه العملية نسجل رقم أطول سطر لأننا سنبدأ به ثم بالسطر الذي يعلوه إلى أن نصل إلى أعلى المخطط، ثم نتابع من السطر الذي تحته حتى نصل إلى نهاية المخطط. و السبب في ذلك هو أننا نريد أن تكون إحداثيات جميع عناصر المخطط موجبة لذلك إذا ضمنا أن عناصر أطول سطر إحداثياتها موجبة فعناصر الأسطر الأقصر منه ستكون موجبة (إن الإحداثيات السالبة تعني أن جزءاً من العنصر يقع خارج منطقة العرض). أما بالنسبة إلى باقي الإجراء فهو مشابه لسابقيه في البرنامجين السابقين.

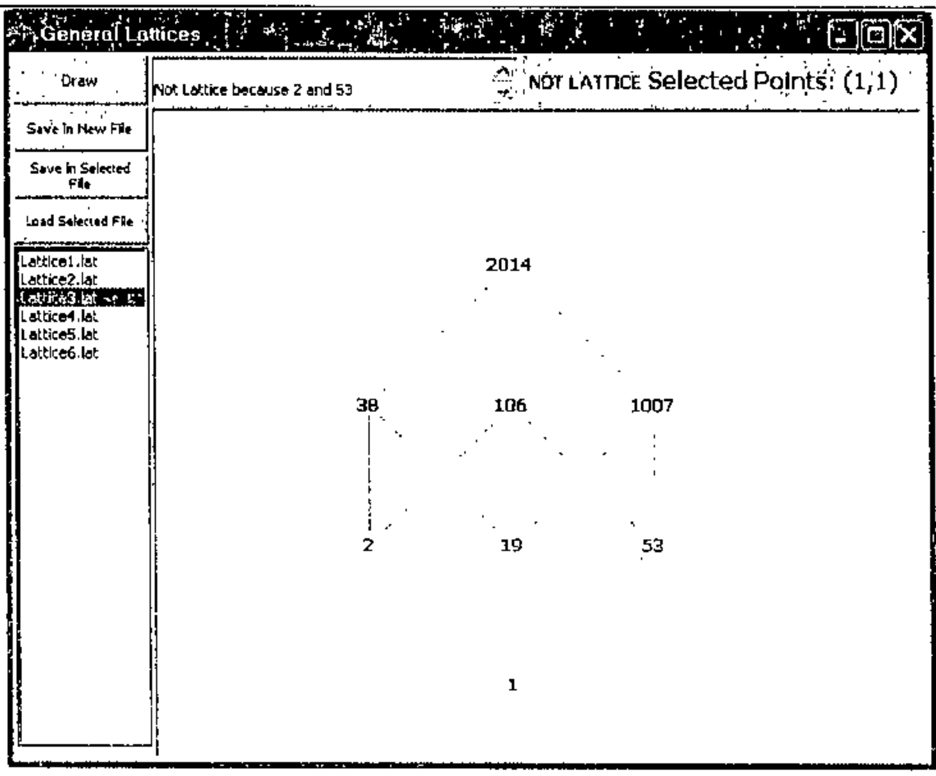
لعل أكثر ما يميز هذا البرنامج بالإضافة إلى ما سبق هو إمكانية حفظ المجموعات المرتبة فيه، و هذه ما سنتناوله مع العديد من الأمور الأخرى في الفصل القادم.

٣. تصميم واجهة البرنامج Designing Interface

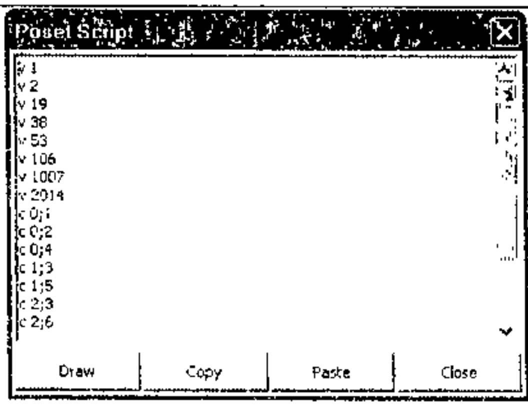
صممنا واجهة هذا البرنامج على نمط سابقيه مع إضافة مربع قائمة ملفات (FileListBox[5]) وعدة أزرار لتحميل وحفظ المجموعة المرتبة في ملف. وذكرنا في التحليل الوظيفي أننا بحاجة إلى نافذة إدخال و تحرير تتألف من مربع نص كبير ضبطنا خاصية تعدد الأسطر له بالقيم صحيح لأننا سنستعمله لإدخال اللغة الوصفية للمجموعة المرتبة المدروسة.

يقوم الزر Draw عند النقر عليه بعرض نافذة الإدخال و التحرير مما يُمكّن المستخدم من إدخال مجموعة مرتبة جديدة أو تحرير المجموعة المرتبة الحالية.

أمّا الأزرار Save in New File، Save in Selected File، Load Selected File فتختص بحفظ المجموعة المرتبة في ملف جديد أو في الملف المحدد أو تحميل الملف المحدد. في أعلى النافذة يوجد مربع نص يعرض الأخطاء التي نحصل عليها من الصنف clsOrderedSet والتي تتضمن سبب أن المجموعة المرتبة ليست شبكة أو عدم إمكانية تعريف علاقة يغطي بين عنصرين كما رأينا عند الحديث عن الأصناف البرمجية.



الشكل ١٦: النافذة الرئيسة



الشكل ١٧: نافذة الإدخال و التحرير

أما نافذة الإدخال و التحرير فبالإضافة إلى مربع النص فهي تحوي أربعة أزرار هي Draw، Copy، Paste، Close وظائفها على الترتيب هي رسم المجموعة المرتبة و دراستها، نسخ اللغة الوصفية إلى الحافظة، لصق محتويات الحافظة في مربع النص، إغلاق النافذة.

٤. كتابة الوحدة البرمجية المساعدة Typing The Module

نستعمل الوحدة البرمجية لكتابة إجراءات عامة والإعلان عن متغيرات على مستوى البرنامج، كما أننا سنضع إجراء حفظ المجموعة المرتبة و إجراء تحميلها فيها.

وفيما يلي نورد النص البرمجي للوحدة modGeneral.

الشيفرة البرمجية الكاملة للوحدة modGeneral

```
Option Explicit

Public DrawFlag As Boolean ' هل تم النقر على زر الرسم في نافذة الإدخال
Public LatScript As String ' اللغة الوصفية للمجموعة المرتبة المدروسة

Public Point1 As Long  ' النقاط المحددة لتحديد شبكة جزئية
Public Point2 As Long

Public Poset As clsOrderedSet ' كائن المجموعة المرتبة

' الإجراء الخاص بتهيئة كائن المجموعة المرتبة بناء على اللغة الوصفية
Public Sub InitPoset()
Dim Lines() As String, i As Integer, p As Long
Dim t As String, Params() As String
' نقسم اللغة الوصفية إلى أسطر
Lines = Split(LatScript, vbNewLine)
LatScript = ""
Set Poset = New clsOrderedSet ' ننشئ كائن مجموعة مرتبة جديدا
Point1 = 0
Point2 = 0
' نكرر من أول سطر إلى آخر سطر
For i = 0 To UBound(Lines)
    ' نستخلص التعليمة و الوسائط
    p = InStr(Lines(i), " ")
    If p Then
        t = Left(Lines(i), p - 1)
        Lines(i) = Mid(Lines(i), p + 1)
    Else
        t = ""
    End If
    Params = Split(Lines(i), ";")
    ' ما هي التعليمة
    Select Case t
    Case "v" ' تعليمة تعريف عنصر
        If UBound(Params) = 0 Then
            ' أضف العنصر إلى كائن المجموعة المرتبة
```

```
            Poset.AddVertex Trim(Params(0))
            LatScript = LatScript & "v " & Trim(Params(0)) _
                                         & vbNewLine
        End If
    Case "c" ' تعليمة تعريف وصلة
        If UBound(Params) = 1 Then
            ' نحاول إضافة الوصلة و نختبر النتيجة
            If Poset.AddConnection(Trim(Params(0)), _
                                   Trim(Params(1))) = False Then
                ' فشلت العملية أعرض الخطأ على النافذة الرئيسة
                frmMain.txtErrors.SelStart = _
                                   Len(frmMain.txtErrors.Text)
                frmMain.txtErrors.SelText = vbNewLine _
                                       & Poset.LastError
            End If
        End If
        LatScript = LatScript & "c " & Trim(Params(0)) & ";" _
                              & Trim(Params(1)) & vbNewLine
    End Select
Next
End Sub

' اجراء اختبار هل هي شبكة و عرض النتيجة على النافذة الرئيسة
Public Sub CheckLattice()
If Poset.IsLattice Then
    ' هي شبكة
    frmMain.lblIsLattice.Caption = "LATTICE"
    frmMain.lblIsLattice.ForeColor = &HC000&
Else
    ' ليست شبكة
    frmMain.lblIsLattice.Caption = "NOT LATTICE"
    frmMain.lblIsLattice.ForeColor = vbRed
    ' أعرض السبب
    frmMain.txtErrors.SelStart = Len(frmMain.txtErrors.Text)
    frmMain.txtErrors.SelText = vbNewLine & Poset.LastError
End If
End Sub

' إجراء حفظ المجموعة المرتبة ضمن ملف
Public Sub SaveFile(f As String)
Dim i As Integer

On Error GoTo ErrH
' نفتح الملف للكتابة
Open App.Path & "\" & f For Output As #1
' نكتب اللغة الوصفية
Print #1, LatScript
Print #1, "END"
' نكتب النقطتين المحددتين
Print #1, Point1 & ";" & Point2
' نكتب إحداثيات العناصر
For i = 0 To frmMain.lbl.Count - 1
    Print #1, frmMain.lbl(i).Left & ";" & frmMain.lbl(i).Top
Next
' نغلق الملف
Close #1

ErrH:
End Sub
```

```
' الإجراء الخاص بتحميل المجموعة المرتبة من ملف
Public Sub LoadFile(f As String)
Dim t As String
' إذا لم يكن الملف موجودا أخرج
If Dir(App.Path & "\" & f) = "" Then Exit Sub
' افتح الملف للقراءة
Open App.Path & "\" & f For Input As #1
' اقرأ اللغة الوصفية
LatScript = ""
Do Until EOF(1)
    Line Input #1, t
    If t = "END" Then Exit Do
    LatScript = LatScript & t & vbNewLine
Loop
' نستدعي إجراء تهيئة المجموعة المرتبة بناء على اللغة الوصفية
InitPoset
' نولد مخطط هاس للمجموعة المرتبة
Poset.GeneratLattice frmMain.lin, frmMain.lbl
' نستدعي إجراء اختبار و عرض نتيجة هل هي شبكة
CheckLattice

Dim v
' نقرأ النقطتين المحددتين
Line Input #1, t
v = Split(t, ";")

Point1 = v(0)
Point2 = v(1)

Dim i As Integer
' نقرأ إحداثيات العناصر و نحركها
For i = 0 To frmMain.lbl.Count - 1
    Line Input #1, t
    v = Split(t, ";")
    frmMain.lbl(i).Move v(0), v(1)

    Dim j As Integer
    ' نحرك الخطوط لتتصل إلى المواضع الجديدة للعناصر
    For j = 0 To frmMain.lin.Count - 1
        If Left(frmMain.lin(j).Tag, Len(i & ";")) = i & ";" Then
            frmMain.lin(j).X1 = frmMain.lbl(i).Left + _
                                frmMain.lbl(i).Width / 2
            frmMain.lin(j).Y1 = frmMain.lbl(i).Top + _
                                frmMain.lbl(i).Height / 2
        ElseIf Right(frmMain.lin(j).Tag, Len(";" & i)) = ";" & i Then
            frmMain.lin(j).X2 = frmMain.lbl(i).Left + _
                                frmMain.lbl(i).Width / 2
            frmMain.lin(j).Y2 = frmMain.lbl(i).Top + _
                                frmMain.lbl(i).Height / 2
        End If
    Next
Next
' أغلق الملف
Close #1
End Sub
```

Some Results of The Program ٥. بعض نتائج البرنامج

إن هذا البرنامج يعطي إمكانية دراسة أية مجموعة مرتبة جزئياً لمعرفة كونهـا شـبكة أم لا ولإنشاء مخططها، ولقد قمنا باختباره على العديد من المجموعـات المرتبـة التـي ليست بالضرورة شبكات.

و نورد في ما يلي بعض النتائج التي حصلنا عليها:

- الشكل ١٨ يمثل شبكة مؤلفة من عشرة عناصر قمنا بتحسينها بسحب العناصر ثم حددنا فيها الشبكة الجزئية [8, 7].

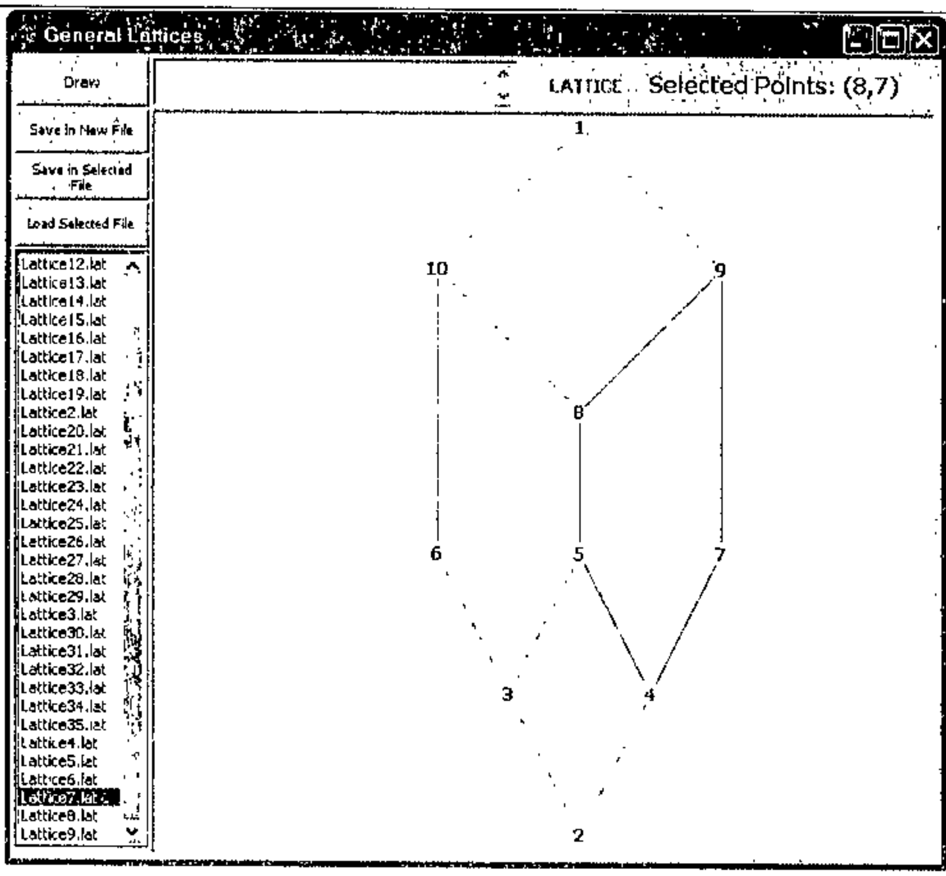
- يمثل الشكل ١٩ المجموعة المرتبة الناتجة عن $(D(2014), \subseteq)$ بعد حذف العلاقة بين العنصرين 106، 2014 مما ينفي عنها صفة الشبكة.

- الشكل ٢٠ يمثل شبكة مجردة مكونة من عشرين عنصراً و ضمنها شبكة جزئية محددة.

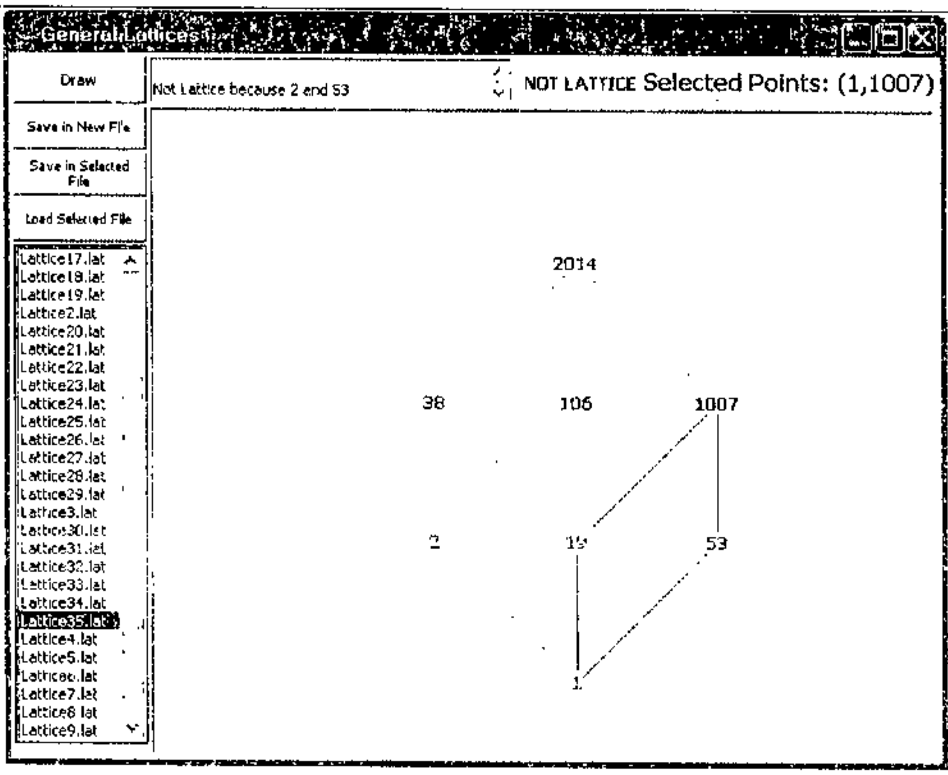
- يمثل الشكل ٢١ الشبكة $(D(12), |)$ بعد إضافة علاقة بين العنصرين 3، 4 مما ينفى عنها صفة الشبكة و يظهر السبب في مربع النص في أعلى النافذة. كما يظهر في الشكل أنها لا يمكن تحديد الشبكة الجزئية [6, 4].

- يظهر الشكل ٢٢ المثالي [d] في شبكة مكونة من العناصر: $\{0, b, c, d, e, f, g, h, 1\}$.

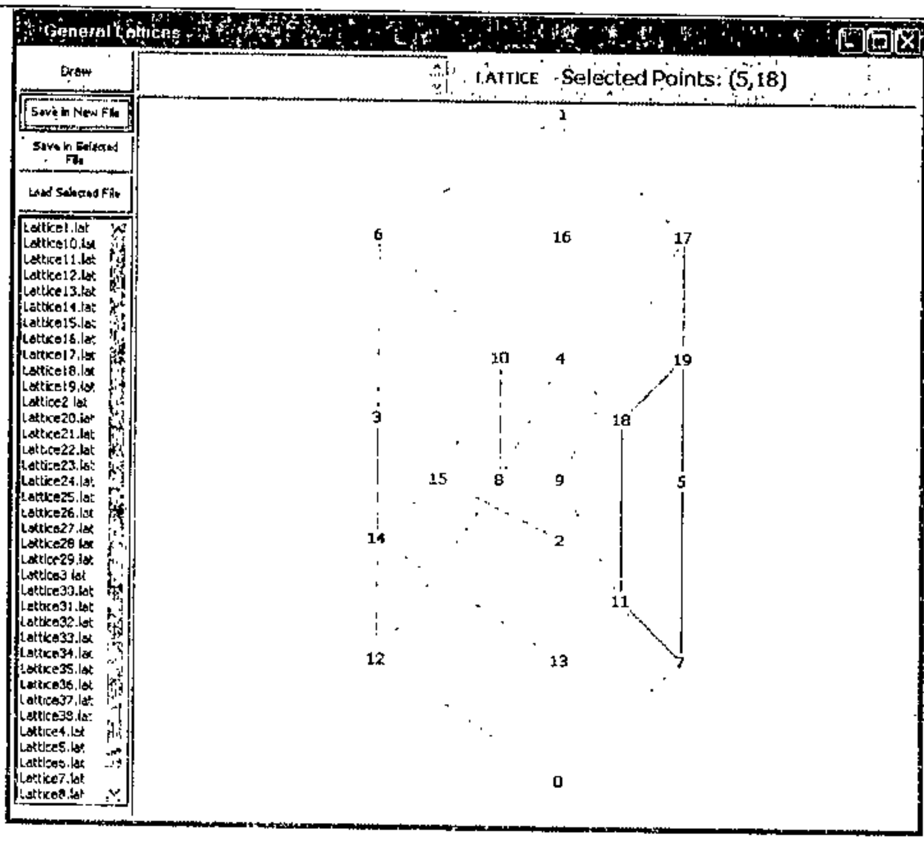
إذاً يمكّن هذا البرنامج من دراسة مخطط أية مجموعة مرتبة جزئيا بغض النظر عن طبيعة عناصرها أو طبيعة علاقة الترتيب المعرفة عليها.



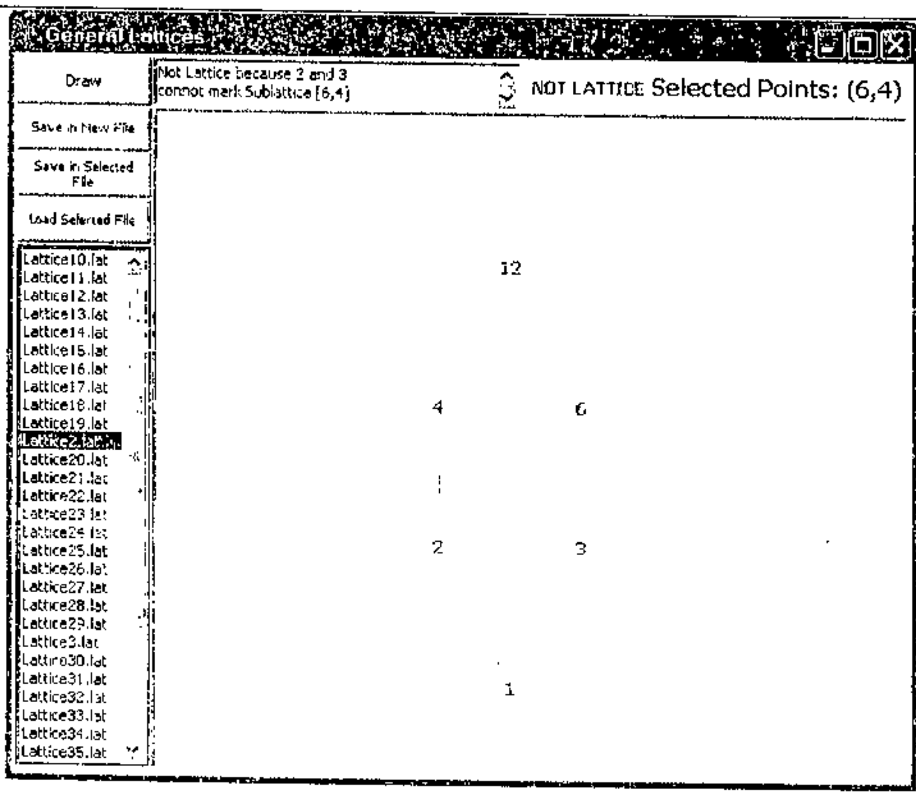
الشكل ١٨



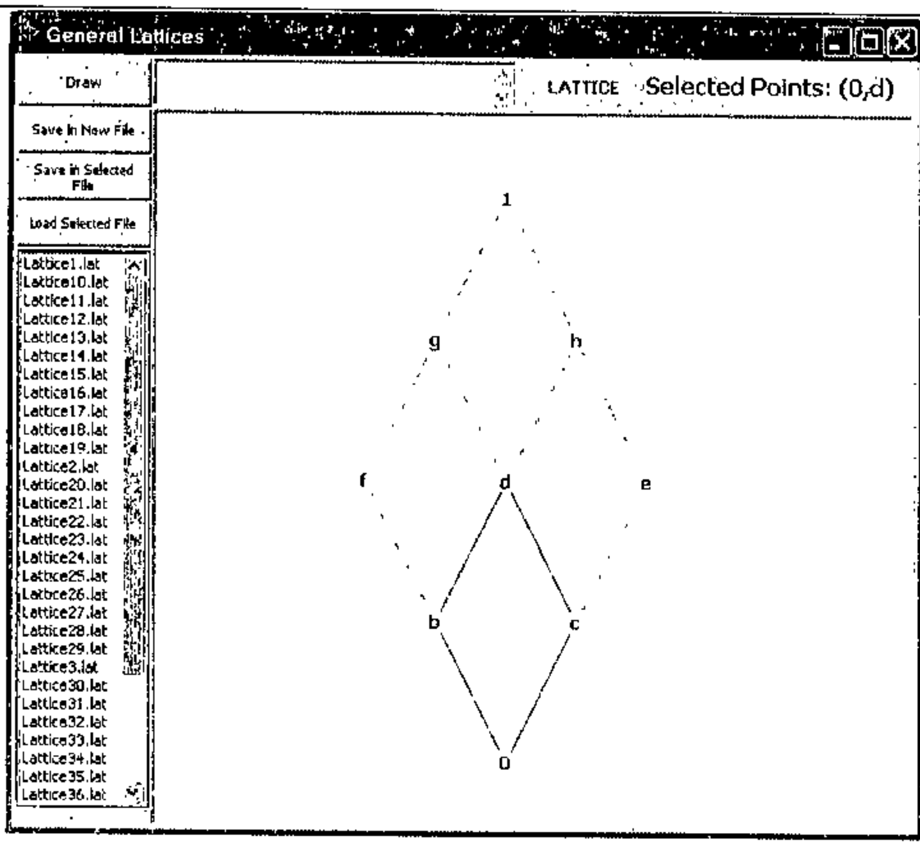
الشكل ١٩



الشكل ٢٠



الشكل ٢١



الشكل ٢٢

الخاتمة

Conclusion

قدمنا في هذا البحث ثلاثة تطبيقات برمجية، التطبيق الأول مختص بدراسة شبكات القواسم والثاني مختص بشبكات المجموعات الجزئية في مجموعة منتهية، أما التطبيق الثالث فمختص بدراسة المجموعات المرتبة المنتهية.

هذه التطبيقات مجتمعة تشكل حزمة متكاملة من البرامج التي تساعد الباحث والمدرس والطالب في دراسة أساسيات نظرية الشبكات مع توفير الجهد و الوقت اللازمين لذلك، كما أنها تشكل لبنة أولى في بناء يجب استكماله، حتى يغدو بالإمكان دراسة جميع البنى الجبرية المجردة على اختلاف أنواعها باستخدام الحاسوب.

Future Works الأعمال المستقبلية

لعل أهم تطوير يمكن إجراؤه هو كتابة خوارزمية تحسين لشكل المخطط تقوم تلقائياً بتحريك العناصر إلى موضعها المناسب من دون الحاجة إلى سحب العناصر يدوياً، مع إضافة ميزة نسخ المخطط الناتج كصورة أو طباعته مما يتيح الاستفادة منه في بحوث أخرى.

كما أن تحسين واجهة البرنامج و إمكانية حفظ عدد شبكات في ملف واحد و إضافة تعليقات أو نصوص معينة عليها من شأنه أن يزيد من فائدة البرنامج.

يبقى أن نذكر أنه يمكن كتابة خوارزميات تتحقق من كون الشبكة معيارية أو توزيعية أو تامة. أو لإيجاد عناصر خاصة كالذرات أو العناصر غير القابلة للاختزال، مما يرفع دراسة البنى الجبرية باستخدام الحاسوب إلى آفاق جديدة.

Software Applications on Lattices

Abstract

These Thesis relate to the study of Divisors Lattices and Lattices of Subsets to create a Hasse Diagram for it and to select its ideals and Filters, and it also studies Posets to determine whether it is a Lattice or not, and to select the ideals and Filters. All of these are done through appropriate software applications to study Lattices .

We wrote for this research three integrated software applications, the first is responsible of the creating the diagram of divisors lattices, so the user can modify the shape of the diagram by dragging elements, and can view ideals, filters and sublattices with different color .

The Second application is a program that is similar in appearance and functions to the Divisors Lattices program, however, it's specializes in Subsets lattices. In this program we solved the problem of representation of sets and the relationship of containment by using simple logical operations .

The Third application is a comprehensive study of any Set in order to learn whether it's a Lattice or not. In order to finish this program we have proven a couple of theories to find the sup(x,y) and inf(x,y) for any two elements x, y in the Poset by calculating the intersections of subsets in the a given Poset:

Theory 1. Let $S_{x,y} = U(x,y) \cap (\bigcap_{z \in U(x,y)} (z\,])$ then $S_{x,y} = \varnothing$ or S have single element Sup(x,y).

Theory 2. Let $N_{x,y} = L(x,y) \cap (\bigcap_{z \in L(x,y)} [\,z))$ then $N_{x,y} = \varnothing$ or N have single element Inf(x,y).

Where U(x,y), L(x,y) the upper bounds set and the lower bounds set of x and y, Consequently.

Syrian Arab Republic
Tishreen University
Faculty of Science
Department of Mathematics

Software Applications on Lattices

Thesis For Master Degree in Mathematics

By
Abdulkader Sheikh Ibrahim

Supervision By
Prof. Dr. Ahmad Algosain

1430
2009